(12)

NUWC-NPT Technical Document 10,212
27 September 1993

Adaptive Control of Telerobotic Systems Worn by Humans

J. B. Bagley Jr.
Combat Control Systems Department

Naval Undersea Warfare Center Division
Newport, Rhode Island

PREFACE

This report, a modified version of the author's thesis submitted in partial fulfillment of the requirements for a master of science degree in electrical engineering at the University of Rhode Island, was prepared under NUWC Project No. 622T30/622T50, "NUWC MSEE Onsite Training Program."

The technical reviewer for this report was Professor Richard Vaccaro of the University of Rhode Island.

Appreciation and gratitude is gratefully extended to Professor Vaccaro for his instruction and for serving as thesis advisor. His insight, comments, and wisdom were invaluable during the preparation of this report.

Reviewed and Approved: 27 September 1993

P. A. La Brecque
Head, Combat Control Systems Department

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	27 September 1993	Final

4. TITLE AND SUBTITLE
Adaptive Control of Telerobotic Systems Worn by Humans

5. FUNDING NUMBERS

6. AUTHOR(S)
J. B. Bagley Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Undersea Warfare Center Division
1176 Howell St.
Newport, RI 02841-1708

8. PERFORMING ORGANIZATION REPORT NUMBER
TD 10,212

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A distinguishing feature of man is his ability to invent mechanical devices to augment his physical capabilities. Traditionally, these machines have been master-slave devices with man controlling the activities of the machine. Recent work has been done to develop a control architecture for the coupled man-machine systems; however, these systems rely on a complete and detailed description of the device and its operating environment. The necessity of an environmental model places limitations on the machine's ability to perform different tasks.

This report discusses how environmental dependence can be overcome through the use of a parameter estimator, which operates on data contained in the system. This report first verifies the performance of the control architecture. A linear estimation technique is then used to allow the system to adapt to an unknown time-invariant environment. Also discussed is how the results of the adaptations can be extended to the problem of an unknown time-varying environment.

14. SUBJECT TERMS
Telerobotic Systems
Adaptive Control
Combat Control Systems

15. NUMBER OF PAGES
92

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

ADAPTIVE CONTROL OF TELEROBOTIC SYSTEMS WORN BY HUMANS

1. INTRODUCTION

The focus of a substantial amount of research in the field of robotics has been in the areas of autonomous systems and remotely teleoperated systems. These areas represent two popular approaches to a class of applications where man is incapable of performing a given task due to his physical limitations. The autonomous or remotely operated machines provide a more capable means for accomplishing the task; however, these two approaches have their own inherent limitations. A great deal of ability is required to create an autonomous robot that can operate in an unstructured environment. This limitation as well as the problems encountered with response requirements and the creation of situation awareness hinder teleoperated systems.

A third approach to this class of problems, combining results of initial investigations into robotics systems worn by humans with the fields of adaptive control and system identification, is discussed.[1] A system is produced that harnesses the versatility of adaptive control to perform dissimilar physical tasks that humans are unable to achieve. This approach lays the foundation for putting the abilities of man back "into the loop" as opposed to trying to remove him.

The concept of a device used to increase the strength of a human operator using a master-slave system with the operator maintaining supervisory control has existed since the early 1960s.[2] Early work performed at the Cornell Aeronautical Laboratory for the Department of Defense starting in 1962 was directed at assessing the feasibility of such a master-slave system. The result of this initial investigation determined that it was impractical to duplicate the full range of human motion and that this type of system would be limited by the physical size of the components required to construct it.[2]

General Electric (GE) Company also investigated the man-amplifier concept between 1966 and 1971.[2] GE designed a master-slave system, known as Hardiman, which consisted of an inner exoskeleton that was coupled to its human host; its motions were controlled by the movements of the host. This linkage provided the master control for a hydraulic outer exoskeleton which acted as the slave. This system advanced to the prototype phase where work ceased.

A more recent investigation has been performed by H. Kazerooni, at the University of Minnesota, who has written a series of papers on a nonmaster-slave approach to the concept of a man-amplifier.[1-5] Kazerooni's concept, known as "Extenders," provides a starting point for the work described in this report. Extenders, as defined by Kazerooni, are a class of robot

manipulators used to extend human arm strength through the transfer of both power and information signals.[1]

Traditional master-slave systems employ a second set of actuators to provide force feedback to the operator. These actuators are also involved in the production of the master control signals used to drive the slave. An operator is never in direct contact with his environment and relies on the duplicate actuators for sensory input. An alternative approach, proposed by the extender concept, eliminates the need for this second set of actuators. The extender senses the force exerted on it by its host and senses the forces which it exerts on the environment. These two signals are used to control the dynamics of the extender. The extender controllers are designed to maintain a relationship between these forces, while the human provides overall control for the extender/host system. Kazerooni states, "Force reflection occurs naturally in the extender system, because the contact forces between the human and the extender let the human feel a scaled-down version of the actual environmental forces on the extender."[1]

A characteristic of this architecture is that a model of the extender's operating environment is incorporated into its controller design imposing a dependence on the extender's capability to operate in an unstructured environment. Modern adaptive control techniques incorporate estimation algorithms to identify unknown system parameters in the absence of a fully specified system model.[6] These techniques can also apply to a system model with inaccurate parameters. The application of parameter estimation allows one to extract parameter information from the data contained within a system. Adding an estimation algorithm to the extender architecture would allow the extender to adapt to an unknown or unstructured environment.

Experimental verification of Kazerooni's work is described in section 2 of this report, which includes a derivation of the system expressions for the components of his architecture, the experimental state space forms used to implement a test system, and simulation data provided to verify Kazerooni's results. This information leads to a description of the case where there is a mismatch between the environmental model and the actual environment. Section 3 introduces the least squares estimation technique as it applies to the environmental mismatch problem. A recursive least squares estimator is then specified and introduced into the overall architecture. Analysis is provided to describe the overall system's performance in the presence of an unknown time invariant environment. Section 4 deals with the applicability of this approach to the time varying parameter case. A bounded gain forgetting (BGF) estimator is specified and added to the original architecture. Analysis is provided to assess the performance of this system. The report concludes with a summary of results and a discussion of topics related to future work.

2. VERIFICATION OF EXTENDER ARCHITECTURE

This section describes the procedures used to verify the extender architecture, which consists of a transfer function description of the models for each of the components comprising the extender system. State space implementations for each subsystem are also provided for the experimental setup used for verification. Differences between the design and experimental configurations are presented, and a description of the composite system is then discussed. Simulation data are used to verify the performance of the basic system. An interactive simulation for dynamic nonlinear systems, called Simnon, is used to verify the extender architecture. The validation of the extender architecture provides a basis for expanding its capabilities. Each of the following descriptions approaches the modeling of a portion of the extender system from an input/output standpoint, placing less importance on the internal structures of each component.

2.1 DEVELOPMENT OF THE EXTENDER MODEL

The dynamic behavior of the extender itself is the first subsystem to be described. Figure 1 shows the block diagram for the extender dynamics. Block G represents the primary compensator for the extender and accepts a control input U, which is generated by the extender controllers. This block consists of the models for the extender's physical hardware and either a position or velocity compensator that governs the motion of the extender based on the input U.

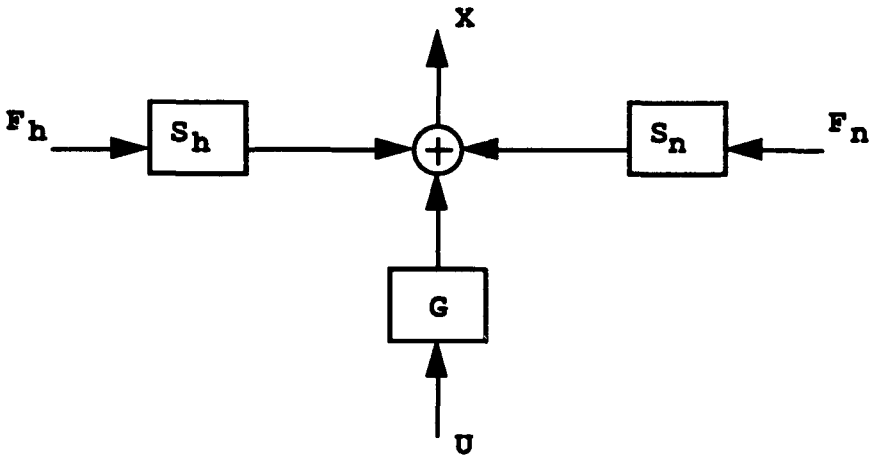


Figure 1. Extender Dynamics

The S_h block represents the model for the extender's sensitivity to forces exerted on it by the human and maps these forces F_h into the output motion of the extender. Sensitivity of the extender to forces generated by interaction with the environment is represented by block S_n, which maps these environmental forces F_n into extender motion. The sum of the outputs from the plant dynamics G and the two force mapping models S_h and S_n results in the output position X of the extender. Equation 1 describes this relationship:

$$X = G\,U + S_h\,F_h + S_n\,F_n. \tag{1}$$

The experimental hardware used by Kazerooni to test his extender architecture consisted of a hydraulic rotary actuator connected to two hollow tubes. The forearm of the operator is placed within the inner tube. A piezoelectric force sensor is placed between the inner and outer tubes to measure the forces exerted on the extender by the operator. A second force sensor is placed at the point of contact with the environment to measure the interaction forces of the extender. Figure 2 depicts this configuration.

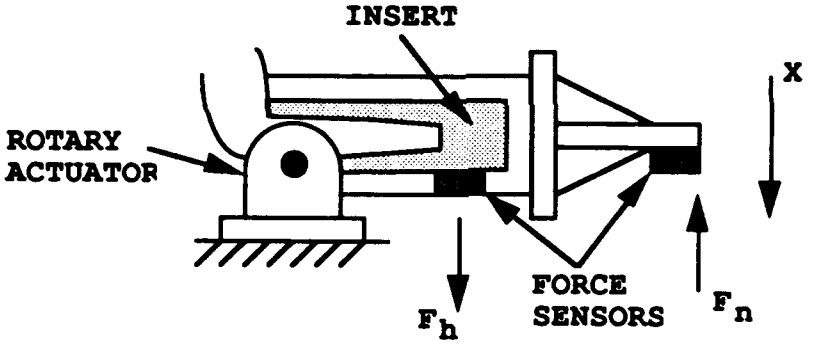


Figure 2. Extender Hardware Setup

The following transfer functions are based on tests performed by Kazerooni with the experimental hardware:[1]

$$G = \frac{X}{U} = \frac{18860.0}{s^3 + 35.55\,s^2 + 1594.25\,s + 18860.0}\ \text{rad/rad}, \tag{2}$$

$$S_h = \frac{F_h}{X} = \frac{1.07\text{x}10^{-2}\,s + 2.53\text{x}10^{-1}}{s^3 + 35.55\,s^2 + 1594.25\,s + 18860.0}\ \text{rad/lbf}, \tag{3}$$

$$S_n = \frac{F_n}{X} = \frac{3.2\text{x}10^{-2}\, s + 7.54\text{x}10^{-1}}{s^3 + 35.55\, s^2 + 1594.25\, s + 18860.0} \text{ rad/lbf.} \quad (4)$$

These transfer functions are then converted into a state space representation used by Simnon. A standard control canonical form is used for these models. The state space relationships are described by the following equations:

$$\dot{x}(t) = \mathbf{A}\, x(t) + \mathbf{B}\, u(t), \quad (5)$$

$$y(t) = \mathbf{C}\, x(t) + \mathbf{D}\, u(t). \quad (6)$$

The values for the matrix **A** and the vectors **B**, **C**, and **D** are derived from the following equations. For a given transfer function H(s),

$$H(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^N + b_1 s^{N-1} + b_2 s^{N-2} + \ldots b_N}{s^N + a_1 s^{N-1} + \ldots a_N},$$

$$\mathbf{A} = \begin{bmatrix} -a_1 & -a_2 & . & . & -a_n \\ 1 & 0 & . & . & 0 \\ 0 & . & . & . & . \\ . & . & . & . & . \\ 0 & . & 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{C} = [b_1 - a_1 b_0 \;\; b_2 - a_2 b_0 \;\; \ldots \; b_n - a_n b_0] \qquad \mathbf{D} = b_0. \quad (7)$$

State space descriptions of the extender dynamics are provided in section 2.5, which discusses the experimental implementation.

2.2 HUMAN ARM DYNAMICS

The modeling of the operators physical dynamics is also included in the architecture of the extender system. This model accounts for the human contact with the extender. Kazerooni assumes that the human arm dynamically behaves as a nonideal source of force interacting with another system.[2] Figure 3 shows the block diagram for the arm dynamics.

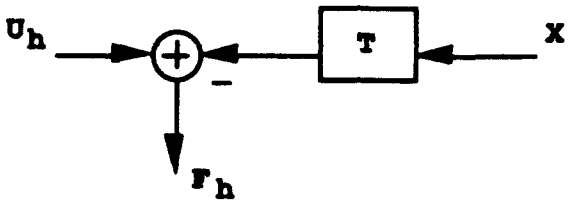


Figure 3. Human Arm Dynamics

The force F_h placed on the extender by the operator is produced from two inputs. The first is U_h representing the information signals which are a result of the humans intent to move his arm. The second is a result of the extender motion X and how this motion is mapped to the contact force. This mapping is accomplished by the transfer function T, which is a measure of human arm sensitivity. The following equation describes this relationship:

$$F_h = U_h - T\,X. \tag{8}$$

An approximate model for the human arm was derived based on experimental data.[1] The model for T was based on both low- and high-frequency experiments conducted at the University of Minnesota, with the following result:

$$T = 2\,(0.143\,s + 1)^2 \text{ lbf/rad.} \tag{9}$$

An alternate expression was used for simulation because equation (9) cannot be realized in canonical state space form. The alternate expression is

$$T = \frac{2\,(0.143\,s + 1)^2\,K^2}{(s + K)^2} \tag{10}$$

where K is an arbitrary constant chosen to place the added poles well into the left half of the s-plane. A value of 100.0 was chosen for K. An expansion of equation (10) gives

$$T = \frac{(4.09\text{x}10^{-2}\,s^2 + 5.72\text{x}10^{-1}\,s + 2)\,K^2}{s^2 + 2Ks + K^2}, \tag{11}$$

which yields a canonical state space representation of

$$\mathbf{A} = \begin{bmatrix} -2\text{x}10^2 & -1\text{x}10^4 \\ 1 & 0 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C = [\,-7.6\times10^{4} \;\; -4.06\times10^{6}] \quad D = 408.9. \tag{12}$$

These values were then implemented in Simnon as the model for the operator's arm.

2.3 OPERATING ENVIRONMENT MODEL

As with the human arm dynamics, a model for the operating environment is also required. For Kazerooni's hardware configuration, an automotive strut was mounted below the extender.[1] The task to be performed was the compression of this strut. The resulting force is measured as the contact force between the extender and its environment. Figure 4 shows the block diagram for the environment dynamics.

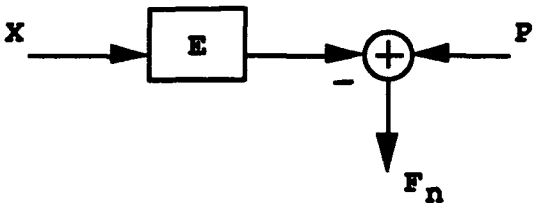


Figure 4. Environment Dynamics

As with the human arm portion of the extender system, there are two sources for environmental force F_n. The first source of this contact force results from the extender motion in which the transfer function E maps this motion to the contact force F_n. The second source of F_n is an external force P applied by the environment on the extender. Throughout all of Kazerooni's work, P is set to zero, meaning that the contact forces sensed by the extender are solely a function of the extender motion. The following equation describes this relationship:

$$F_n = -Ex + P. \tag{13}$$

The automotive strut exhibits a stiffness and damping, which is represented by a first-order model. The inertia of the strut is negligible. Following is an expression for the model of the automotive strut E:

$$E = 200s + 2050 \text{ lbf/rad}. \tag{14}$$

A canonical state space form for E was not used in experimental verification. The implementation of E will be described in the discussion of the composite system detailed in a later section.

2.4 CONTROLLER EQUATIONS

The final component of the extender architecture consists of the controllers which generate the command input for the extender system. Figure 5 shows the block diagram for the extender controllers.

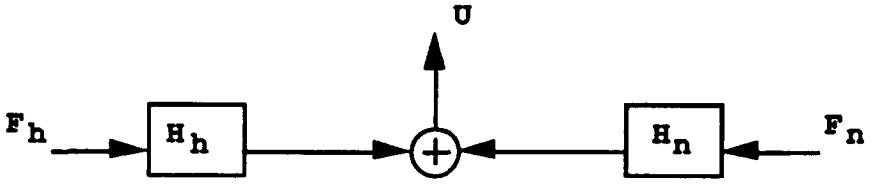


Figure 5. Extender Controllers

There are two controllers in the extender architecture which perform similar tasks: controller H_h maps the operator contact force F_h to the command input U, and controller H_n maps the environmental interaction force F_n to the same command input U. The following equation describes this relationship:

$$U = F_h H_h + F_n H_n. \tag{15}$$

Two different types of controllers are described in the remainder of this section. The first of these controller configurations maintains a desired relationship between the input force F_h and the output force F_n. This configuration results from the derivations in Kazerooni's papers.[1-5] The second controller configuration maintains a desired relationship between the input force F_h and the output position X. This configuration is an additional form resulting from this investigation.

The derivation for the force-force relationship is based on the following desired dynamics:

$$F_n = Q^{-1} F_h \tag{16}$$

where Q^{-1} maps the operator input force F_h to the output force F_n. Following is the derivation of this relationship that starts with equation (1):

$$X = G\,U + S_h\,F_h + S_n\,F_n.$$

If a primary controller G is chosen with a large open-loop gain, and the gains in S_h and S_n are small, then combining equation (1) with equation (15) yields

$$X \cong G\,H_h\,F_h + G\,H_n\,F_n. \tag{17}$$

H_h and H_n are chosen to have the following forms:

$$H_h = -2\,G^{-1}\,E^{-1}\,Q^{-1}, \tag{18}$$

$$H_n = G^{-1}\,E^{-1}. \tag{19}$$

Substituting (18) and (19) into (17) yields

$$X \cong -2\,E^{-1}\,Q^{-1}\,F_h + E^{-1}\,F_n. \tag{20}$$

Assuming that P = 0 for equation (13), equation (20) can be rewritten as

$$-E^{-1}\,F_n \cong -2\,E^{-1}\,Q^{-1}\,F_h + E^{-1}\,F_n. \tag{21}$$

Equation (21) reduces to the target relationship

$$F_h \cong Q\,F_n. \tag{22}$$

The value for Q can be chosen as a constant or transfer function. For the purposes of this report, Q is chosen as a positive constant. Substituting equations (2) and (14) into equations (18) and (19) gives the following transfer function descriptions for the force-force controller case:

$$H_h = \frac{2}{Q}\,\frac{s^3 + 35.55\,s^2 + 1594.25\,s + 18860.0}{18860.0\,(200.0\,s + 2050.0)}, \tag{23}$$

$$H_n = \frac{s^3 + 35.55\,s^2 + 1594.25\,s + 18860.0}{18860.0\,(200.0\,s + 2050.0)}. \tag{24}$$

As with the equations for the human arm dynamics, these two equations are not realizable in canonical form. Two additional poles in the left half of the s-plane are added to each equation yielding

$$H_h = \frac{2}{Q} \frac{(s^3 + 35.55\, s^2 + 1594.25\, s + 18860.0)\, K^2}{18860.0\,(200.0\, s + 2050.0)\,(s + K)^2}, \tag{25}$$

$$H_n = \frac{(s^3 + 35.55\, s^2 + 1594.25\, s + 18860.0)\, K^2}{18860.0\,(200.0\, s + 2050.0)\,(s + K)^2}. \tag{26}$$

A value of 100.0 is chosen for K. Equations (25) and (26) yield the following state space representations:

For H_n,

$$\mathbf{A} = \begin{bmatrix} -210.25 & -12050.0 & -102500.0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{C} = [\, -4.63\text{x}10^{-1} \;\; -27.72 \;\; -221.73] \qquad \mathbf{D} = -2.65\text{x}10^{-3}. \tag{27}$$

Values for H_h are the same as those for H_n with the following changes:

$$\mathbf{C}(H_h) = 2/Q\, \mathbf{C}(H_n) \qquad \mathbf{D}(H_h) = 2/Q\, \mathbf{D}(H_n).$$

The derivation for the force-position form of the extender controllers follows a similar format starting with equation (1) and subsequently equation (17). H_h and H_n are chosen to have the following forms:

$$H_h = 2\, G^{-1}\, R^{-1}, \tag{28}$$

$$H_n = G^{-1}\, E^{-1}. \tag{29}$$

Substituting (28) and (29) into (17) yields:

$$X \cong 2\, R^{-1}\, F_h + E^{-1}\, F_n. \tag{30}$$

Assuming P = 0 in equation (13), and substituting equation (13) into equation (30) results in

$$X \cong 2 R^{-1} F_h - X. \tag{31}$$

Rearranging terms produces the following desired target dynamics:

$$Fh \cong R X. \tag{32}$$

As with Q, R is chosen to be a positive constant. The transfer function form of H_n does not change. Substituting (2) into (28) yields the following transfer function for H_h:

$$H_h = \frac{2}{R} \frac{s^3 + 35.55 s^2 + 1594.25 s + 18860.0}{18860.0}. \tag{33}$$

This equation is not realizable in canonical state space form. Three poles are added to equation (33) yielding

$$H_h = \frac{2}{R} \frac{(s^3 + 35.55 s^2 + 1594.25 s + 18860.0) K^3}{18860.0 (s + K)^3}. \tag{34}$$

In this case, a value of 10.0 was chosen for K. The following canonical state space representation results from equation (34):

$$A = \begin{bmatrix} -30.0 & -300.0 & -1000.0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$C = [\, 2.94 \times 10^{-1} \;\; 68.62 \;\; 946.98] \qquad D = 5.3 \times 10^{-2}. \tag{35}$$

The models described by equations (25), (26), and (34) represent continuous time implementations of the extender controllers. Most of the test results are based on continuous time forms. The extender controllers described by equations (23) and (24) were also discretized using the bilinear transform [7]

$$s = \frac{2}{T} \frac{(z - 1)}{(z + 1)} \tag{36}$$

where T is the discrete sampling time. Applying the bilinear transform to equations (23) and (24) yields the following discrete transfer functions for T = 0.01:

$$H_h(z) = \frac{2}{Q} \frac{1.36\text{x}10^{-7}\, z^3 - 3.5\text{x}10^{-7}\, z^2 + 3.12\text{x}10^{-7}\, z - 9.5\text{x}10^{-8}}{z^3 - 1.57\, z^2 + 0.712\, z - 0.1}, \tag{37}$$

$$H_n(z) = \frac{1.36\text{x}10^{-7}\, z^3 - 3.5\text{x}10^{-7}\, z^2 + 3.12\text{x}10^{-7}\, z - 9.5\text{x}10^{-8}}{z^3 - 1.57\, z^2 + 0.712\, z - 0.1}. \tag{38}$$

Several simulation runs were done to verify the controllers in a discrete form. The results of these runs are presented in a section 2.7.

2.5 EXPERIMENTAL IMPLEMENTATIONS

Kazerooni presents the derivations of his extender architecture in two sections. The first of these deals with the analytical derivation of the extender system description.[2,4] Sections 2.1, 2.2, and 2.3 of this report describe the components of the "design" system. The second portion of Kazerooni's work describes the "experimental" hardware used to verify the design system.[3,5] This section addresses the differences between the analytical design expressions and the experimental hardware setup. The implementation differences between the design discussed previously and the simulation system used to verify the extender architecture are also described. The two areas where differences occur are in the models for the extender dynamics and the environmental dynamics.

The extender hardware model used by Kazerooni to implement the extender architecture differs from the analytical description described in section 2.1. The simplified architecture depicted in figure 1 is replaced by the hardware setup depicted in figure 6. Block G_p corresponds to the open loop transfer function that maps the input current I to the extender position X, which is similar in form to the block G from figure 1. This block is described by the following transfer function:

$$G_p = \frac{553888.75}{s^2 + 35.55\, s + 1560.25} \text{ rad/ampre.} \tag{39}$$

This equation yields the following control canonical state space form:

$$A = \begin{bmatrix} -35.55 & -1560.25 \\ 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C = [0.0 \;\; 553888.75] \qquad D = 0. \tag{40}$$

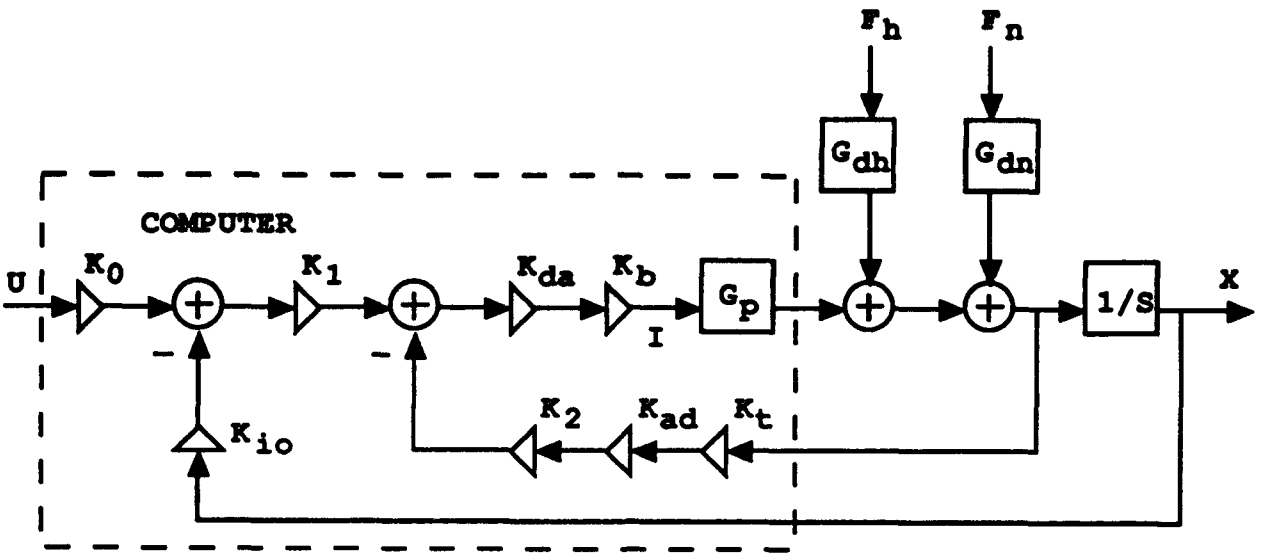


Figure 6. Experimental Extender Dynamics

The block G_{dh} represents the mapping from F_h to the output X, which is similar to the S_h block of figure 1. The transfer function for G_{dh} is

$$G_{dh} = \frac{8.92\text{x}10^{-4}\, s + 2.11\text{x}10^{-2}}{s^2 + 35.55\, s + 1560.25} \text{ rad/(lbf inch).} \qquad (41)$$

Equation (41) has the following state space form:

$$A = \begin{bmatrix} -35.55 & -1560.25 \\ 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C = [\,8.92\text{x}10^{-4} \;\; 2.11\text{x}10^{-2}\,] \quad D = 0. \qquad (42)$$

The block G_{dn} represents the open loop transfer function, which maps F_n to the extender position X. G_{dn} corresponds to S_n in figure 1 and has the following transfer function:

$$G_{dn} = \frac{2.67\text{x}10^{-3}\, s + 6.32\text{x}10^{-2}}{s^2 + 35.55\, s + 1560.25} \text{ rad/(lbf inch).} \qquad (43)$$

Equation (43) yields the following state space form:

$$A = \begin{bmatrix} -35.55 & -1560.25 \\ 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C = [2.67\text{x}10^{-3} \;\; 6.32\text{x}10^{-2}] \quad D = 0. \tag{44}$$

It should be noted that the transfer function descriptions for G_p, G_{dh}, and G_{dn} are one order lower than their corresponding models from figure 1. This factor of 1/S was extracted from each of the design transfer functions and appears in figure 6. Both position and velocity are fed back to provide primary compensation for the extender dynamics. Table 1 lists values for the gains shown in figure 6.

Table 1. Experimental System Gains

Gain	Symbol	Value	Units
Servo Controller Board Gain	K_b	$4.65\text{x}10^{-3}$	amps/volt
Tachometer Gain	K_t	$1.69\text{x}10^{-1}$	volts/(rad/sec)
A to D Converter Gain	K_{ad}	2048/1.25	volts
Encoder I/O Gain	K_{io}	1592.0	number/rad
Precompensator Gain	K_0	1592.0	n/a
Position Gain	K_1	$9.4\text{x}10^{-1}$	n/a
Velocity Gain	K_2	$9.77\text{x}10^{-3}$	n/a
D to A Converter Gain	K_{da}	10/2048	volts

The second area where a difference exists is in the implementation of the environmental model. This difference occurs between the design implementation discussed in section 2.3 and the simulation implementation used to verify Kazerooni's results. The extraction of the 1/S factor from the extender dynamics models allows for a simpler implementation of the transfer function described by equation (14). Figure 7 shows the experimental implementation of the environmental transfer function E, where E_0 represents the environmental stiffness and E_1 represents the environmental damping. These two constants have the corresponding values of 2050 lbf/rad and 200 lbf/(rad/sec). The signal V represents the sum of the outputs from G_p, G_{dh}, and G_{dn}, and the output of the summation corresponds to the output of the environmental model.

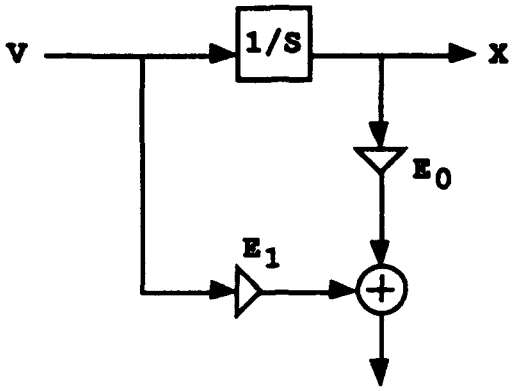


Figure 7. Experimental Environmental Dynamics

2.6 COMPOSITE SYSTEMS

The combination of the components described in the previous sections produces the composite extender system. Two forms of the composite system exist: the design extender composite system and the experimental composite extender system. The first of these represents the design system which is described in figure 8.

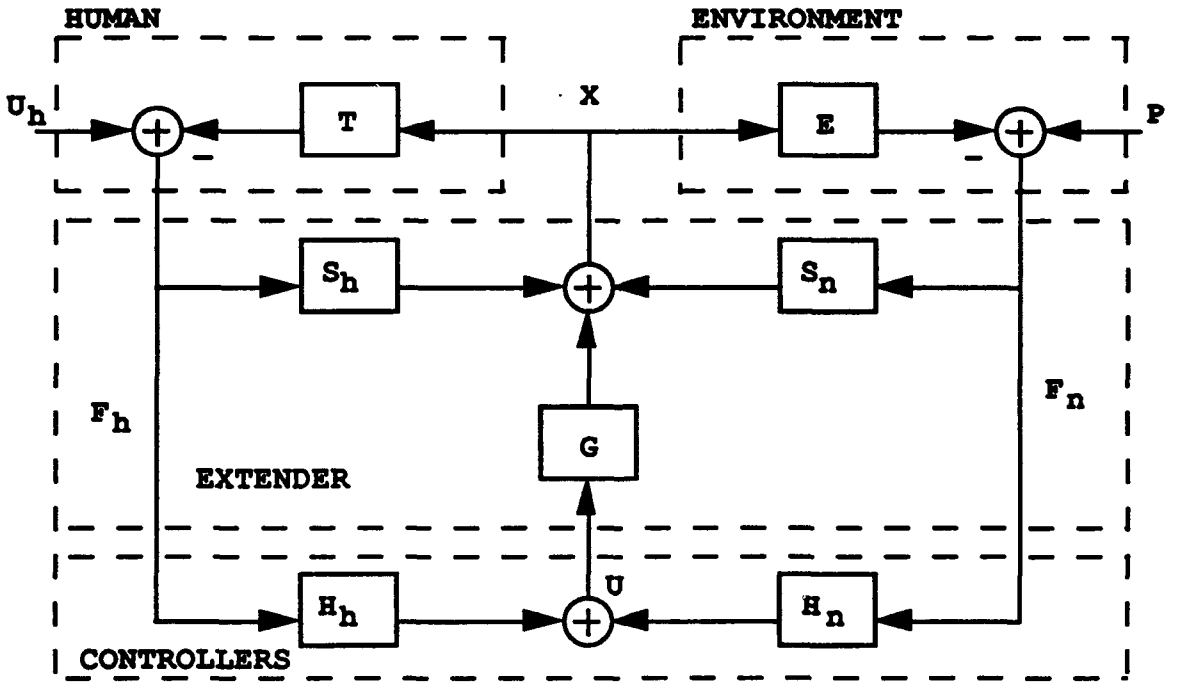


Figure 8. Design Extender Composite System

This figure shows the interconnection of the extender dynamics, environment, human operator and controller models. The relationship of F_h and F_n can be verified using Mason's rule.[8] The resulting overall system transfer function is

$$H(s) = \frac{F_n}{F_h} = \frac{S_h E + H_h G E}{1 + S_n E + H_n G E}. \quad (45)$$

Using the assumptions described for equation (17) and substituting equations (18) and (19) in for H_h and H_n produces:

$$H(s) = \frac{F_n}{F_h} = \frac{-2 Q^{-1}}{2}. \quad (46)$$

Rearranging terms produces the target relation for the overall system transfer function:

$$F_n = -Q^{-1} F_h. \quad (47)$$

The minus sign results from the environmental force acting in the opposite direction from the operator force. This result will also appear in the graphs of the test data for the force-force verification. If the sign of the force is ignored then equation (16) results from the above derivation. The experimental composite system differs from the design composite system (refer to section 2.5). Figure 9 shows the block diagram for the experimental composite system.

2.7 SIMNON SIMULATION RESULTS

Simulations were performed to verify Kazerooni's architecture and results.[1] A dynamic system modeling software product called Simnon was used for these simulations. Initial tests were aimed at verifying the relationship derived for the force-force case. Subsequent simulation runs were also performed to examine the force-position relation and test the use of digital controllers in the baseline system. Simnon simulation code for the composite system is provided in the appendix.

A periodic low-frequency signal was selected by Kazerooni for the input signal U_h in the experimental extender system. The selection of this signal is based on the frequency characteristics of the signals associated with human motion.[2,3] An initial input sinusoid of

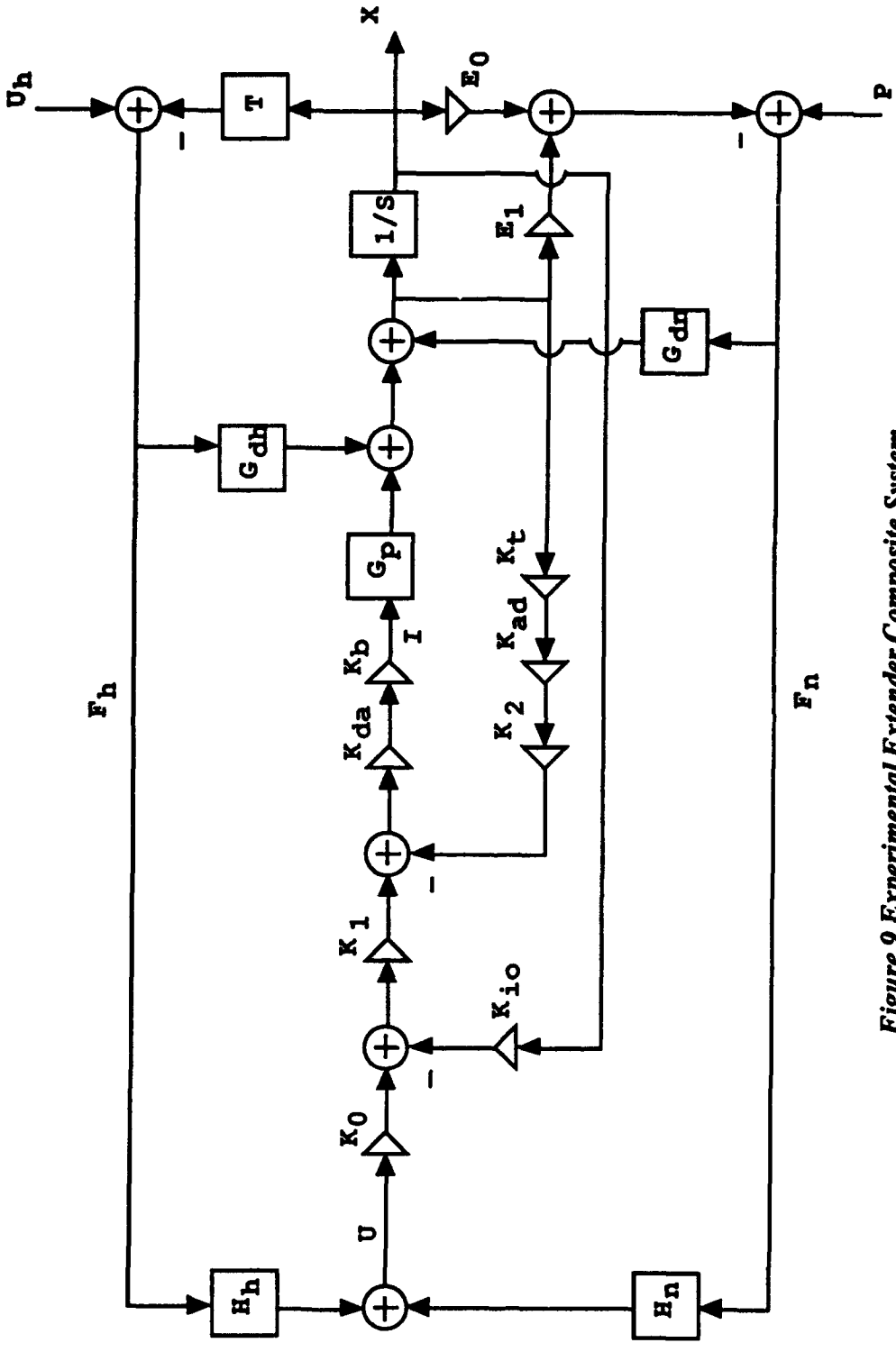


Figure 9 Experimental Extender Composite System

magnitude 5 and period 2π was used for U_h during the initial Simnon simulation runs, and a value of 0.5 was selected for Q from equation (16), which results in a factor of two force amplification. Figure 10 shows the graph of the input force F_h and the output force F_n for the initial Simnon setup. The data presented shows that proportionality between F_h and F_n was achieved for the baseline extender system. This result verifies the extender architecture.

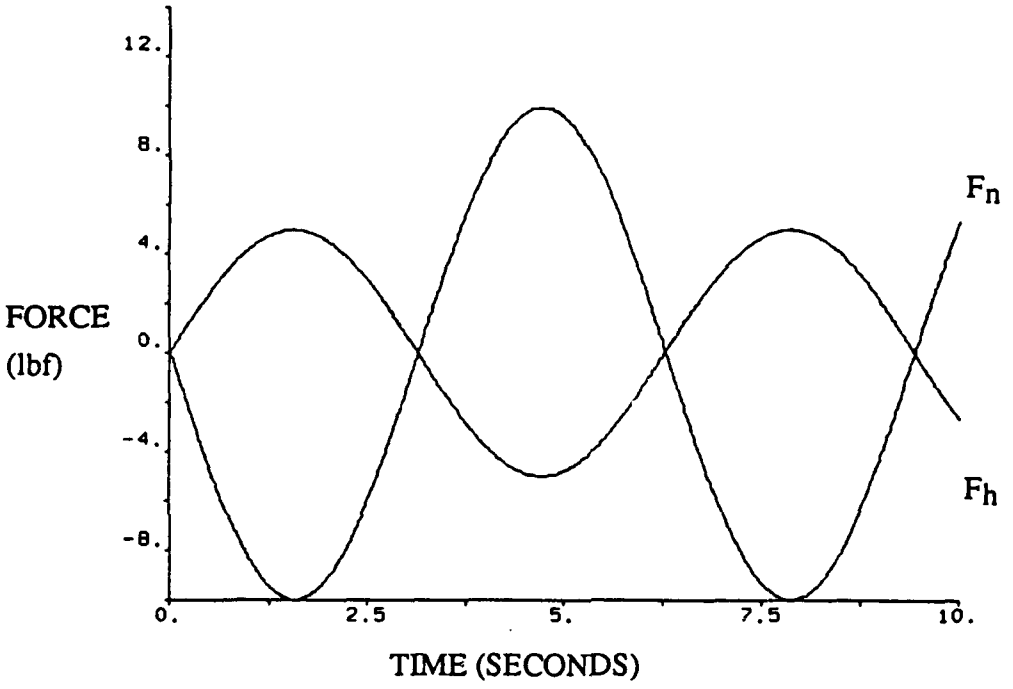


Figure 10. Simnon Force-Force Simulation Results

Simulation runs were also made with different values of Q, ranging from 1.0 to as low as 0.01 producing force amplification factors from 0 to 100. Input frequencies were also varied to test the extender system. Sinusoids with frequencies as high as 1.6 Hz were used to drive the simulation system. Results from these test cases also verified the force relation described by equation (16).

The initial sinusoidal input and amplification factor used to test the force-force relation were also used to test the force-position relation. Figure 11 shows the graph of the input force Fh and the output position X for the Simnon simulation, and the data verify that the target relationship described by equation (32) is maintained.

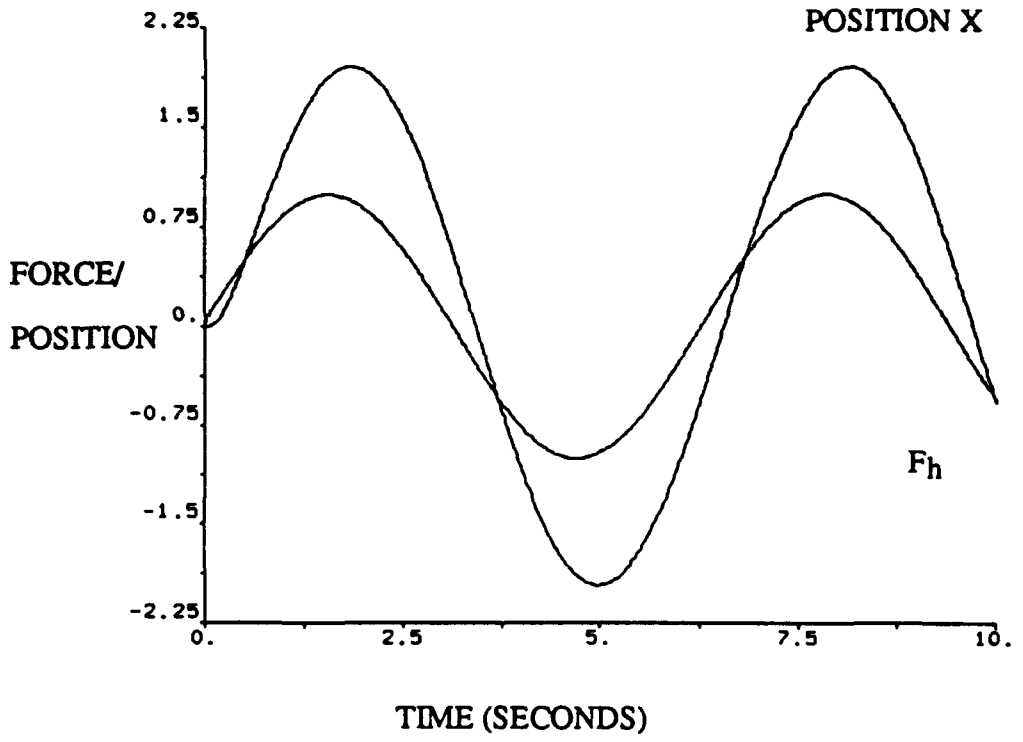


Figure 11. Simnon Force-Position Simulation Results

The baseline extender system was also tested with discrete controllers described by equations (37) and (38) for the force-force relationship. Simnon simulation code for the discrete controllers is provided in the appendix. The discrete controllers were tested with the same force amplification factors and input frequencies as in the continuous time case, and their performance was identical to that of the continuous time controller models. A discrete sampling rate of 0.01 second was chosen for these cases.

The simulation data collected during these simulation runs verify that the extender architecture developed by Kazerooni produces the proportional force amplification of the target relations for both discrete and continuous time controllers. Test data also verify that the extender controllers can be designed to maintain a force-position relationship. The verification of these relationships provides a base upon which further expansion of the extender principle can be achieved.

3. ESTIMATION OF AN UKNOWN TIME INVARIANT ENVIRONMENT

This section focuses on a characteristic of the extender architecture described in section 2. Equations (18) and (19) define the structures on the extender controllers which produce the proportional force dynamics. Both equations contain the term E, which represents the model for the extender operating environment. For the simulations performed to test the extender architecture, E is modeled as a spring damper with constant coefficients.[1] The necessity of having this term in the controller model limits the performance of the extender system to a unique environment. Interaction with another environment would result in different behavior, and the extender controllers would have to be changed to allow for interaction with different environments. The extender would be unable to perform a task if the environment was unknown.

Section 3 addresses the application of parameter estimation to this problem. Implications of an environmental mismatch on extender performance are discussed, the estimation theory is developed, and how it applies to the problem of an unknown environment is discussed. Finally, simulation results are analyzed to assess the effectiveness of parameter estimation in the extender environment.

3.1 ENVIRONMENTAL MISMATCH

Simnon simulations were performed to examine the impact of an environmental model mismatch. Several tests were performed to evaluate extender performance when the controller model for the environment did not match the actual operating environment. Figure 12 shows a plot of two cases of an environmental mismatch, with three curves representing the extender output force F_n. The first curve shows the reference output produced by the matched system discussed in section 2, and the second curve shows the output force when the stiffness and damping of the actual environment are reduced by 20 percent. This environment is described by equation (48):

$$E_1 = 160s + 1640 \text{ lbf/rad.} \tag{48}$$

The third curve shows the output force for a system with a stiffness and damping that have increased by 20 percent. Equation (49) describes this environment:

$$E_1 = 240s + 2460 \text{ lbf/rad.} \tag{49}$$

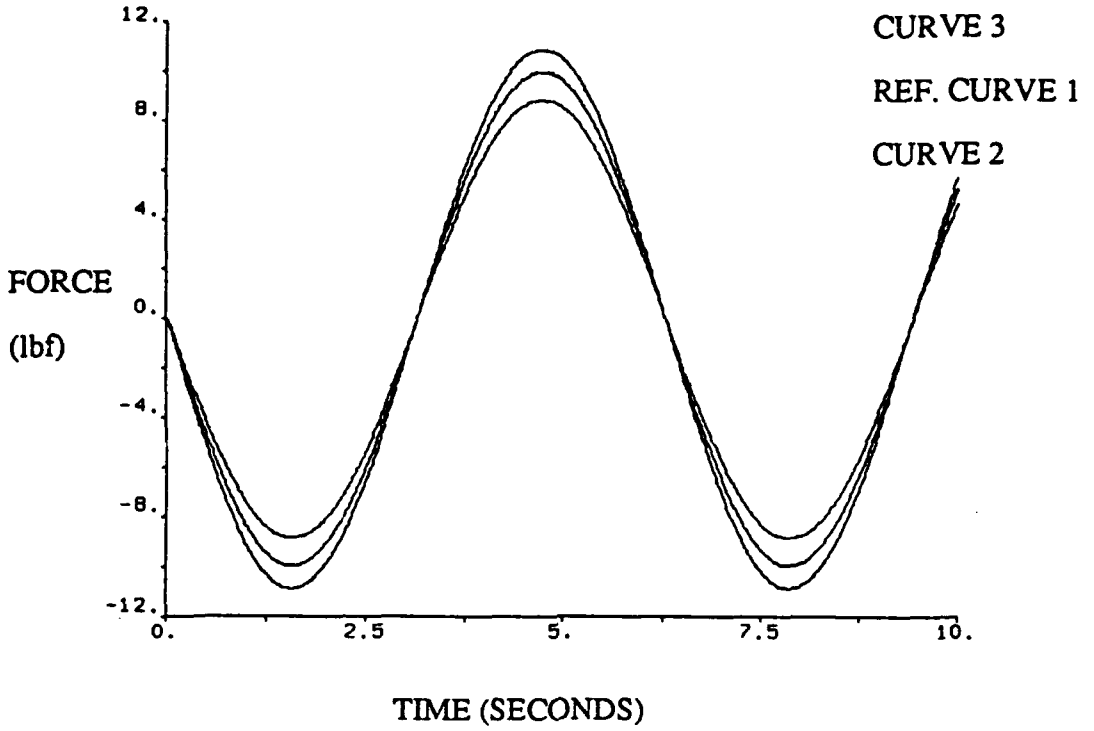


Figure 12. Output Force F_n for Mismatched Environments

The subscripted E_1 refers to the actual environment and E_0 will be used to represent the design environment implemented in the extender controllers in subsequent equations. The data described by the second and third curves show that force amplification is affected by a mismatch in the environmental model. Curve two shows a reduced force amplification factor. Curve three shows an increased force amplification factor. Simulations showed that system response time was also affected by a mismatched environment. Changes in the actual environment caused the output force F_n to lead or lag the output from the original baseline simulation run. Different combinations of stiffness and damping produced varied force factors and lead/lag delays.

The behavior shown by figure 12 can be attributed to a pole-zero cancellation, which occurs when the actual environment matches the design environment. An environmental mismatch produces a pole-zero mismatch in the overall system transfer function. The following derivation describes the pole-zero mismatch relationship starting with the relation shown in equation (17):

$$X \equiv G H_h F_h + G H_n F_n.$$

H_h and H_n are chosen to have the same forms as described by equations (18) and (19) noting that E^{-1} has been replaced by E_0^{-1}.

$$H_h = -2\, G^{-1}\, E_0^{-1}\, Q^{-1} \tag{50}$$

$$H_n = G^{-1}\, E_0^{-1}\,. \tag{51}$$

Substituting (50) and (51) into (17) produces

$$X \equiv -2\, E_0^{-1}\, Q^{-1}\, F_h + E_0^{-1}\, F_n. \tag{52}$$

Assuming $P = 0$, substituting E_1 for E in equation (13), and combining this with equation (52) yields

$$-E_1^{-1}\, F_n \equiv -2\, E_0^{-1}\, Q^{-1}\, F_h + E_0^{-1}\, F_n. \tag{53}$$

Rearranging terms in equation (53) results in the following:

$$\frac{Q}{2}\,\frac{E_0^{-1} + E_1^{-1}}{E_0^{-1}}\, F_n = F_h. \tag{54}$$

Multiplying the left side of equation (54) by E_0/E_0 yields

$$\frac{Q}{2}\,(E_0\, E_1^{-1} + 1)\, F_n = F_h. \tag{55}$$

The resulting transfer function for the extender system is described by equation (56):

$$H(s) = \frac{F_n}{F_h}\,\frac{2}{Q}\,\frac{1}{(E_0\, E_1^{-1} + 1)} \tag{56}$$

where E_0 has the form **a**s + **b** and E_1 has a similar form **c**s + **d**. If the coefficients of these two first-order polynomials match, then equation (56) reduces to the proportional relationship described by equation (22). If these forms are substituted into equation (56) then the following transfer function description results:

$$H(s) = \frac{F_n}{F_h}\,\frac{2}{Q}\left(\frac{c}{c+a}\right)\frac{s+\left(\frac{d}{c}\right)}{s+\left(\frac{b+d}{a+c}\right)}. \qquad (57)$$

Equation (57) describes the first-order system which results when a mismatch occurs in the environmental models. If a match occurs in the environment, then the single pole and zero of equation (57) cancel and H(s) reduces to a constant Q^{-1}. The remainder of this section addresses the application of a parameter estimator to produce values for the actual stiffness and damping of the environment, and these values can then be used to update the extender controllers. This process will match the controller parameters of E_0 to the actual environment modeled by E_1.

3.2 PARAMETER ESTIMATION

A continuous time implementation of a recursive least squares estimator was selected to produce parameter values for the extender environment. These estimated values are based on the input and output signals of the extender system F_h and F_n. This selection was primarily based on the fact that the extender system fits into a linear model for a dynamic system. A continuous time implementation was selected to remain consistent with the baseline extender system. In general, the least squares technique provides relatively smooth convergence and good robustness with respect to noisy data.[6] If a system can be described by the following relation, then a linear estimator can provide information about system's parameters based on available data. This model is expressed as

$$Y(t) = W(t)\,\Theta \qquad (58)$$

where Y(t) is a vector of output data, W(t) is a matrix of known signal data, and Q is a vector of unknown parameters. The standard least squares technique is based on minimizing the integral of the squared error with respect to the unknown parameters of Q. This error is expressed as

$$J = \int_0^t \| Y(t) - W(t)\,\hat{\Theta}(t) \|^2\, dt \qquad (59)$$

where $\hat{\Theta}$ is the predicted value of Θ. The minimization of equation (59) produces the batch form of the estimator in matrix form:

$$\hat{\Theta} = \left(\int_0^t W(t)^T W(t)\, dt\right)^{-1} \int_0^t W(t)^T Y(t)\, dt, \tag{60}$$

which is written recursively producing the following [6,9]:

$$\frac{d\Theta}{dt} = -P(t)\, W(t)^T (\hat{Y}(t) - Y(t)), \tag{61}$$

$$\frac{dP}{dt} = -P(t)\, W(t)^T W(t)\, P(t)\,. \tag{62}$$

P(t) is a matrix of gains used to weight the measurement signals W(t). Equations (61) and (62) provide the continuous time relations used to update the estimates of the terms in Θ. The error expressed as $\hat{Y}(t) - Y(t)$ provides a measure of the convergence of the parameter estimates. The next section will apply these general expressions to the extender system.

3.3 EXTENDER PARAMETER ESTIMATOR DERIVATIONS

Section 3.2 described the principles used to derive the expressions for the estimates of the extender environmental parameters of E_1. The derivation begins with the transfer function development described in section 3.1. The formulation in section 3.1 resulted in a first-order transfer function for the mismatched extender system described by equation (57), which is based on the analytical expressions for the extender controllers. Experimental expressions for the extender controllers contain two additional poles that were introduced to make the controllers realizable in canonical state space form. The derivations from section 3.1 will be repeated to account for this difference starting with equation (17). The following are the forms of the experimental controllers:

$$H_h = (-2\, G^{-1}\, E_0^{-1}\, Q^{-1}) \frac{K^2}{(s + K)^2}\,, \tag{63}$$

$$H_n = (G^{-1}\, E_0^{-1}) \frac{K^2}{(s + K)^2}\,. \tag{64}$$

Substituting (63) and (64) into equation (17) yields

$$X \cong (-2\,E_0^{-1}\,Q^{-1}\,F_h + E_0^{-1}\,F_n)\frac{K^2}{(s+K)^2}. \tag{65}$$

Assuming P = 0 for equation (13), equation (65) is rewritten as

$$-E_1^{-1}\,F_n \cong (-2\,E_0^{-1}\,Q^{-1}\,F_h + E_0^{-1}\,F_n)\frac{K^2}{(s+K)^2}. \tag{66}$$

Rearranging terms in equation (66) results in the transfer function for the experimental extender system:

$$H(s) = \frac{F_n}{F_h} = \frac{2}{Q}\,\frac{1}{(E_0\,E_1^{-1}\,\frac{(s+K)^2}{K^2} + 1)}. \tag{67}$$

Substituting **as + b** for E_0 and **cs + d** for E_1 in equation (67) and multiplying out the terms of the denominator results in

$$H(s) = \frac{2K^2c}{Qa}\,\frac{s+\frac{d}{c}}{s^3 + (2K+\frac{b}{a})\,s^2 + (\frac{2Kb}{a} + K^2 + \frac{K^2c}{a})\,s + \frac{K^2b}{a} + \frac{K^2d}{a}}. \tag{68}$$

The third-order equation described by equation (68) is then rewritten as follows to simplify notation:

$$H(s) = \frac{Y}{U} = k0\,\frac{s+z0}{s^3 + a1\,s^2 + a2\,s + a3}$$

where

$$k0 = \frac{2K^2c}{Q\,a} \qquad z0 = \frac{d}{c} \quad a1 = 2\,K + \frac{b}{a} \quad a2 = \frac{2Kb}{a} + K^2 + \frac{K^2c}{a} \quad a3 = \frac{K^2b}{a} + \frac{K^2d}{a}. \tag{69}$$

Variables $k0$, $z0$, $a2$, and $a3$ are the unknown parameters to be estimated. The variable $a1$ is a function of known constants. Equation (68) describes the transfer function that is then filtered and placed in the desired form described by equation (58). Notation that is used to describe the filtering operation makes reference to both time domain and frequency domain data.[6] The

filtering operation starts with multiplying both numerator and denominator of equation (69) by $1/l_f^3$ where l_f is a known constant. This multiplication is performed to provide unity scaling in the filters. The resulting transfer function is expressed as a ratio of two polynomials in s.

$$H(s) = \frac{Y(s)}{U(s)} = \frac{B(s)}{A(s)}$$

where

$$B(s) = \frac{k0}{\lambda_f^3} s + \frac{k0\, z0}{l_f^3}$$

$$A(s) = \frac{1}{\lambda_f^3} s^3 + \frac{a1}{\lambda_f^3} s^2 + \frac{a2}{\lambda_f^3} s + \frac{a3}{\lambda_f^3}. \tag{70}$$

At this point, the known polynomial is introduced:

$$A0(s) = (s + \lambda_f)^3 = s^3 + 3\lambda_f s^2 + 3\lambda_f^2 s + \lambda_f^3. \tag{71}$$

Equation (70) can be rewritten in the following form:

$$A(s)\, Y(s) = B(s)\, U(s). \tag{72}$$

Multiplying both sides of (72) by the filtering operation $\lambda_f^3/A0(s)$ results in

$$\frac{\lambda_f^3\, A(s)}{A0(s)} Y(s) = \frac{\lambda_f^3\, B(s)}{A0(s)} U(s). \tag{73}$$

Separating out the term Y in equation (73) results in

$$Y(s) = \frac{A0(s) - \lambda_f^3\, A(s)}{A0(s)} Y(s) + \frac{\lambda_f^3\, B(s)}{A0(s)} U(s). \tag{74}$$

Equation (74) describes the third-order extender system in terms of the output Y, signal data consisting of the filtered input U and filtered output Y, and unknowns Θ in A(s) and B(s) where

$$W(s) = \left[\frac{s^2\, Y(s)}{A0(s)} \; \frac{s\, Y(s)}{A0(s)} \; \frac{Y(s)}{A0(s)} \; \frac{s\, U(s)}{A0(s)} \; \frac{U(s)}{A0(s)} \right],$$

$$\Theta = \left[(3\lambda_f - a_1)\ (3\lambda_f^2 - a_2)\ (\lambda_f^3 - a_3)\ k_0\ k_0 z_0\right]^T. \tag{75}$$

The first product of this system of five terms is known. Moving this product to the left of equation (74) reduces the system to four terms, resulting in a new system equation:

$$Y_f(s) = W(s)\,\Theta \tag{76}$$

where Y_f has the following form:

$$Y_f(s) = \left[\ 1 - \frac{(3\lambda_f - a_1)\,s^2}{A_0(s)}\ \right] Y(s)\,. \tag{77}$$

An inverse laplace transform can be applied to equation (76) to produce the time domain form. W and Θ are expressed in their final form with a factor of λ_f^3 moved from Θ to W:

$$W(s) = \left[\ \frac{\lambda_f^3\, s\, Y(s)}{A_0(s)}\ \frac{\lambda_f^3\, Y(s)}{A_0(s)}\ \frac{\lambda_f^3\, s\, U(s)}{A_0(s)}\ \frac{\lambda_f^3\, U(s)}{A_0(s)}\ \right]$$

$$\Theta = \left[\left(\frac{3}{\lambda_f} - \frac{a_2}{\lambda_f^3},\ 1 - \frac{a_3}{\lambda_f^3},\ \frac{k_0}{\lambda_f^3},\ \frac{k_0 z_0}{\lambda_f^3}\right)\right]. \tag{78}$$

Equation (78) describes the extender system in the form discussed in section 3.2. The terms of the vector Θ in equation (78) can then be used to arrive at formulas which map the four unknowns $\Theta_{1\text{-}4}$ to the environmental parameters **c** and **d** of equation (57). Solving the first term in Θ for **c** yields

$$c = \frac{3\lambda_f^2\, a}{K^2} - \frac{2b}{K} - a - \frac{\Theta_1\, a\, \lambda_f^3}{K^2}\ . \tag{79}$$

The third term in Θ can also be solved for **c**:

$$c = \frac{\Theta_3\, Q\, a\, \lambda_f^3}{2K^2}. \tag{80}$$

The second term in Θ is solved for **d** yielding

$$d = \frac{\lambda_f^3 a}{K^2} - b - \frac{\Theta_2 a \lambda_f^3}{K^2}. \tag{81}$$

Finally, the fourth term in Θ can also be solved for **d** producing

$$d = \frac{\Theta_4 Q a \lambda_f^3}{2K^2}. \tag{82}$$

Equations (79) through (82) give four formulas to arrive at two unknowns. To maximize the information provided by each of the results, they are combined into a matrix representation:

$$\begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \Theta_3 \\ \Theta_4 \end{bmatrix} = \begin{bmatrix} \frac{-K^2}{a\lambda_f^3} & 0 \\ 0 & \frac{-K^2}{a\lambda_f^3} \\ \frac{2K^2}{Q a \lambda_f^3} & 0 \\ 0 & \frac{2K^2}{Q a \lambda_f^3} \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} + \begin{bmatrix} \frac{3}{\lambda_f} - \frac{2K b}{a\lambda_f^3} - \frac{K^2}{\lambda_f^3} \\ 1 - \frac{K^2 b}{a\lambda_f^3} \\ 0 \\ 0 \end{bmatrix} \tag{83}$$

$$Y = W \Theta + N.$$

This matrix equation can then be solved in the least squares sense yielding a single expression for the unknowns **c** and **d** in terms of $\Theta_{1\text{-}4}$. The following series of equations reduce equation (83) to these expressions.

$$W' = (W^T W)^{-1} = \begin{bmatrix} W_1 & 0 \\ 0 & W_1 \end{bmatrix}, \tag{84}$$

$$W'' = W' W^T = \begin{bmatrix} W_2 & 0 & W_3 & 0 \\ 0 & W_2 & 0 & W_3 \end{bmatrix}, \tag{85}$$

$$\begin{bmatrix} c \\ d \end{bmatrix} = W'' Y_t \tag{86}$$

where

$$W_1 = \frac{Q^2 a^2 \lambda_f^6}{K^4 Q^2 + 4K^4}, \qquad W_2 = \frac{-Q_2 a \lambda_f^3}{K^2 Q^2 + 4K^2}, \qquad W_3 = \frac{2Q a \lambda_f^3}{K^2 Q^2 + 4K^2}$$

$$Y_t = \Theta - N = \begin{bmatrix} \Theta_1 - \frac{3}{\lambda_f} + \frac{2K\,b}{a\,\lambda_f^3} + \frac{K^2}{\lambda_f^3} \\ \Theta_2 - 1 + \frac{K^2\,b}{a\,\lambda_f^3} \\ \Theta_3 \\ \Theta_4 \end{bmatrix},$$

$$c = W_2 \left(\Theta_1 - \frac{3}{\lambda_f} + \frac{2K\,b}{a\,\lambda_f^3} + \frac{K^2}{\lambda_f^3} \right) + W_3\,\Theta_3, \tag{87}$$

$$d = W_2 \left(\Theta_2 - 1 + \frac{K^2\,b}{a\,\lambda_f^3} \right) + W_3\,\Theta_4 . \tag{88}$$

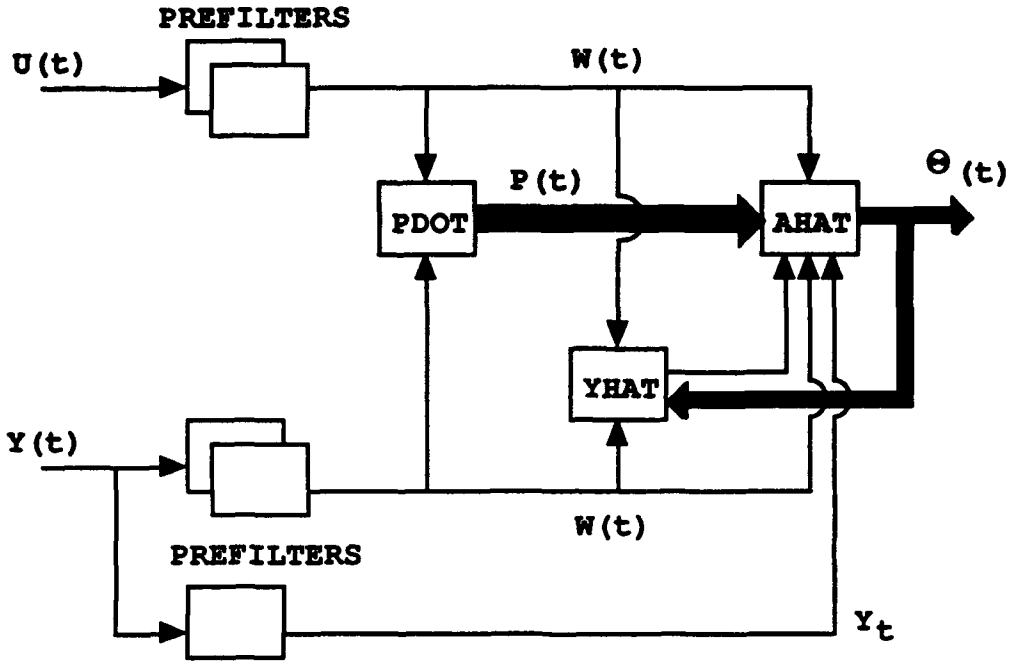
Equations (87) and (88) represent the final results of the derivation for the parameter estimator equations. The equations throughout this derivation provide the basis for the Simnon simulation implementation of the parameter estimator provided in the appendix. Results produced in this section will be applied to the baseline extender system to produce a new composite adaptive system. A description of this new architecture and the results of simulation runs using this technique are discussed in sections 3.4 and 3.5.

3.4 COMPOSITE ESTIMATION SYSTEM

The formulas described in section 3.3 provided a detailed description for the implementation of the environmental parameter estimator that starts with prefilters described by equations (77) and (78). Equation (77) describes the form for the output prefilter, which produces the filtered output $Y_t(t)$. The four terms in the expression for W from equation (78) describe the forms of four signal prefilters that produce filtered input signals from U(t) and Y(t). These filtered inputs and filtered outputs are then used by three models that implement the actual parameter estimator. The first of these models updates the gain matrix P(t), and its implementation is based on equation (62). Then, the output of the gain model is used by the second model, which produces the estimates for the four unknown parameters $\Theta_{1\text{-}4}$. This parameter update model is an implementation of equation (61). The third model uses the parameter estimates $\hat{\Theta}$ produced by the parameter update model to generate the error term described by the following equation:

$$\text{Parameter Error} = \hat{Y}(t) - Y(t) = W(t)\,\hat{\Theta} - Y(t). \tag{89}$$

This parameter error is then fed back into the parameter update model. The interconnection of the prefilters and estimator models is shown in figure 13.



GAIN UPDATE MODEL	PDOT
PARMETER UPDATE MODEL	AHAT
PARAMETER ERROR MODEL	YHAT

Figure 13. Parameter Estimator Implementation

The parameter error provides a weighting function, which is used in producing the parameter estimates. Figure 13 shows the structure that was first tested as a standalone system to evaluate the estimator implementation and behavior. This standalone testing was performed in two phases. The first phase of testing was intended to verify the estimator in a configuration independent of the extender system. In this phase of testing, a linear combination of input signals was fed into the estimator system with no prefiltering. These tests were aimed at verifying the three core models of the estimator implementation. W(t) consisted of a vector of the input signals, Θ was a vector of integer constants, and Y(t) was produced from the product of W(t) and Θ^T. These signals were fed into the estimator which produced accurate values for $\hat{\Theta}$. Different combinations of input signals and Θ values were used during this phase of testing. In each case, accurate estimates for Θ were produced.

The second phase of testing expanded on the results of the first phase. As in the first phase of testing, the input signals W(t) and output Y(t) were fabricated and fed into the estimator system. The goal of these tests was to verify the complete estimator system in the presence of a simplified extender model. Values for the coefficients of the actual and design environments are substituted into equation (68) to produce a third-order transfer function, which was then used to produce the output Y(t) based on a given input U(t).

Initial tests identified two important characteristics of the least squares estimator. The first of these addresses the idea of persistent excitation. The input signal used to drive a system should contain sufficient information to produce parameter convergence.[6] In the case of the simplified extender model, the input signal required sufficient frequency content to excite the modes of the extender system. A sum of low-frequency sinusoids was chosen to drive the simplified extender model instead of using a single sinusoidal input signal. This sum provided a level of excitation which produced quick parameter convergence.

The second characteristic of the least squares estimator is the selection of an initial value for the gain matrix P(t). P(0) was chosen as a diagonal matrix with the non-zero terms chosen to be as large as possible to produce quick convergence without being influenced by noise.[6] A higher initial gain also produced larger overshoot in the parameter convergence. The only limitations encountered in the second phase of testing were imposed by the numerical limits of the Simnon simulator when calculating the matrix products of equation (62).

A variety of simulation runs were performed to test the prefiltered estimator with the simplified extender model. During these runs, different combinations of the unknown environmental coefficients were used in generating the input and output signals. Table 2 presents parameter estimate data for six of these test cases. The tests used to produce the data in table 2 were based on a controller environment $E_0 = 200s + 2050$, and an initial gain P(0) = 10,000. Simulation runs were 2 seconds long. The data in table 2 show that the parameter estimator described in section 3.2 converges to predicted values when driven by the simplified extender model. Based on this performance, the full adaptive system can be tested and verified.

Table 2. Unit Test Simulation Results

E_1		$\Theta_1 x10^{-3}$		$\Theta_2 x10^{-1}$		$\Theta_3 x10^{-2}$		$\Theta_4 x10^{-1}$	
c	d	Pre	Act	Pre	Act	Pre	Act	Pre	Act
200	1845	7.95	7.94	8.0525	8.003	4.0	3.99	3.69	3.78
220	2050	6.95	6.93	7.95	7.947	4.4	4.4	4.1	4.11
180	1845	8.95	8.83	8.0525	7.876	3.6	3.57	3.69	4.11
220	1845	6.95	6.94	8.0525	8.045	4.4	4.4	3.69	3.7
180	2255	8.95	9.09	7.8475	7.912	3.6	3.59	4.51	4.37
200	2050	7.95	7.99	7.95	8.01	4.0	4.0	4.1	3.99

Pre => Predicted values
Act => Simulated values

3.5 FULL SYSTEM ANALYSIS

The final series of simulations tested the full adaptive extender architecture in the presence of an unknown time invariant environment. The estimation system verified in section 3.4 was connected to the extender system described in section 2. Figure 14 illustrates the complete composite adaptive extender system and shows that F_h and F_n are used as the input and output signals for the estimator system. The output of the estimator Θ is sampled by the discrete system DCE, which extracts the final estimates for the environmental coefficients. The discrete system DCE is described by equations (83)-(88). This update is triggered by a threshold value of the parameter error. The coefficient estimates are fed into the controllers, which have been modified to accept new parameter values, the controllers are updated, and the extender system maintains the force-force relation based on these new parameters.

A number of simulations were performed to evaluate the composite system. Values for the environmental coefficients, estimator gain, and input sinusoids were varied to examine estimator performance. The composite system tests lead to similar conclusions concerning the estimator setup. A persistently exciting input signal is required to provide sufficient information for parameter convergence. As in the unit test case, a signal with frequency content within the bandwidth of the composite system produced better convergence. The estimator gain also affected parameter convergence. An increased gain produced faster convergence, and increased overshoot as the parameters converged. Table 3 provides parameter estimate data for six of the simulation tests. Initial values and inputs are based on the characteristics described above.

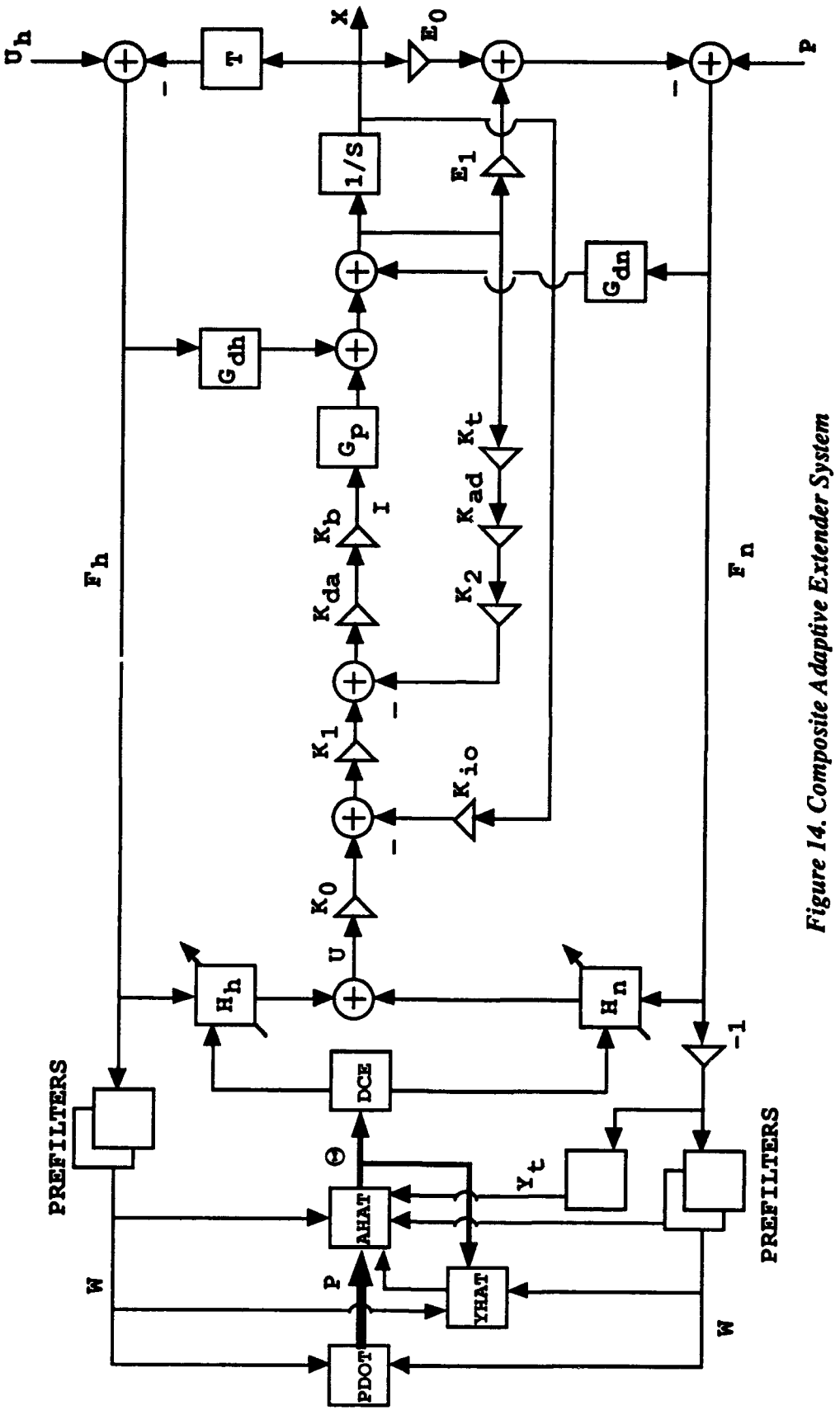


Figure 14. Composite Adaptive Extender System

Table 3. Full System Simulation Results

E_1		$\Theta_1 x10^{-3}$		$\Theta_2 x10^{-1}$		$\Theta_3 x10^{-2}$		$\Theta_4 x10^{-1}$	
c	d	Pre	Act	Pre	Act	Pre	Act	Pre	Act
200	1845	7.95	8.01	8.0525	8.056	4.0	3.96	3.69	3.63
220	2050	6.95	6.96	7.95	8.073	4.4	4.38	4.1	3.84
180	1845	8.95	8.86	8.0525	7.813	3.6	3.55	3.69	4.13
220	1845	6.95	6.92	8.0525	8.096	4.4	4.38	3.69	3.6
180	2255	8.95	9.17	7.8475	7.903	3.6	3.55	4.51	4.38
200	2050	7.95	8.06	7.95	7.991	4.0	3.96	4.1	4.01

Pre => Predicted values
Act => Simulated values

The simulations that produced the data in table 3 had initial estimator gains set to 6000 with the initial controller environment coefficients set to $E_0 = 200s + 2050$. Simulations were 2 seconds long. Figure 15 shows plots of the parameter convergence for the first case of table 3. Plot 15e shows the error of the estimator as the four parameters Θ_{1-4} converge to their final values. At time 0.45, the estimator error has steadied at a level below the error threshold, at which point the routine DCE is triggered. The values of Θ are taken and new values are calculated for the environmental coefficients c and d. These values are then sent to the controllers H_h and H_n, which update their models. Figure 16 shows the parameters c and d as they converge to their final values. The data presented in table 3 and figures 15 and 16 show that the least squares parameter estimator converges to the correct results for the environmental coefficients of E_1, and this validates the idea that the extender system can be modified to allow it to adapt to an unknown time invariant environment. The process of changing controller parameters is automated, which enables the extender to interact with a varied environment and with an environment where the model is unknown. This process greatly increases the abilities of the extender system.

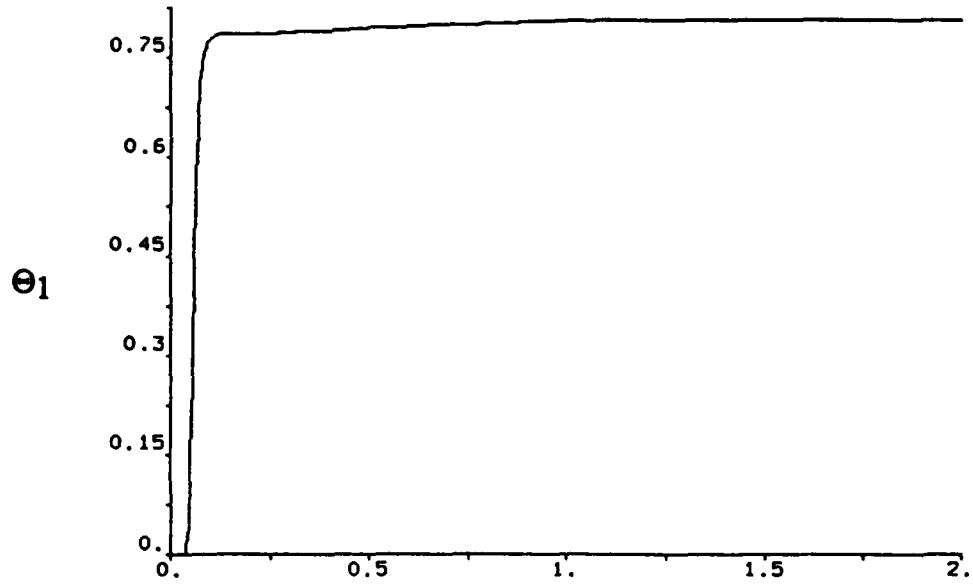


Figure 15a. Convergence of Θ_1

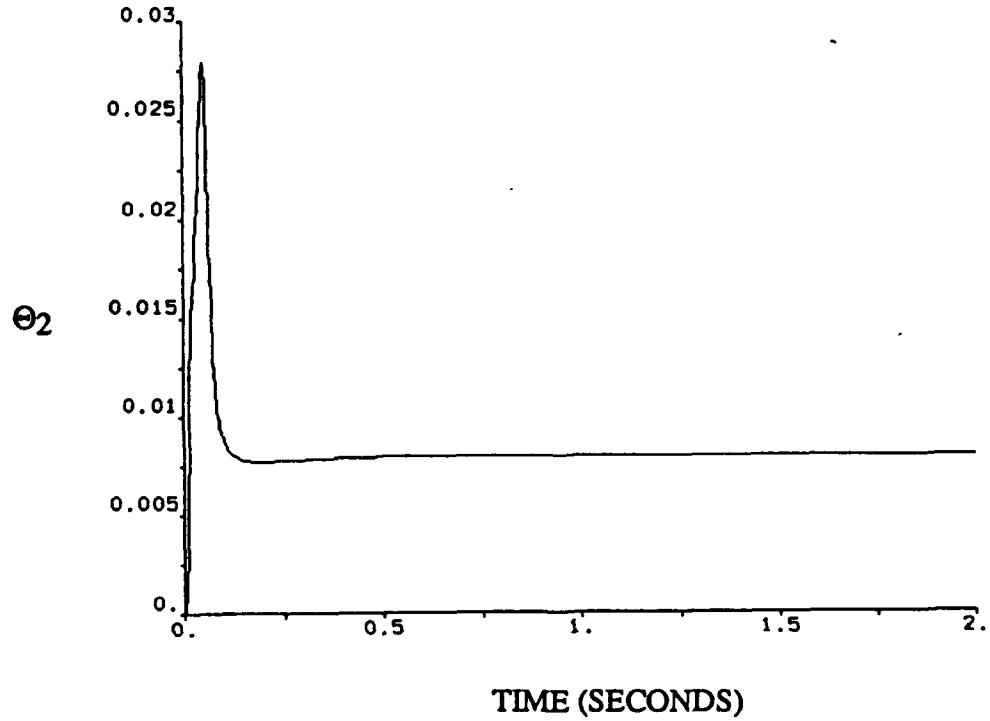


Figure 15b. Convergence of Θ_2

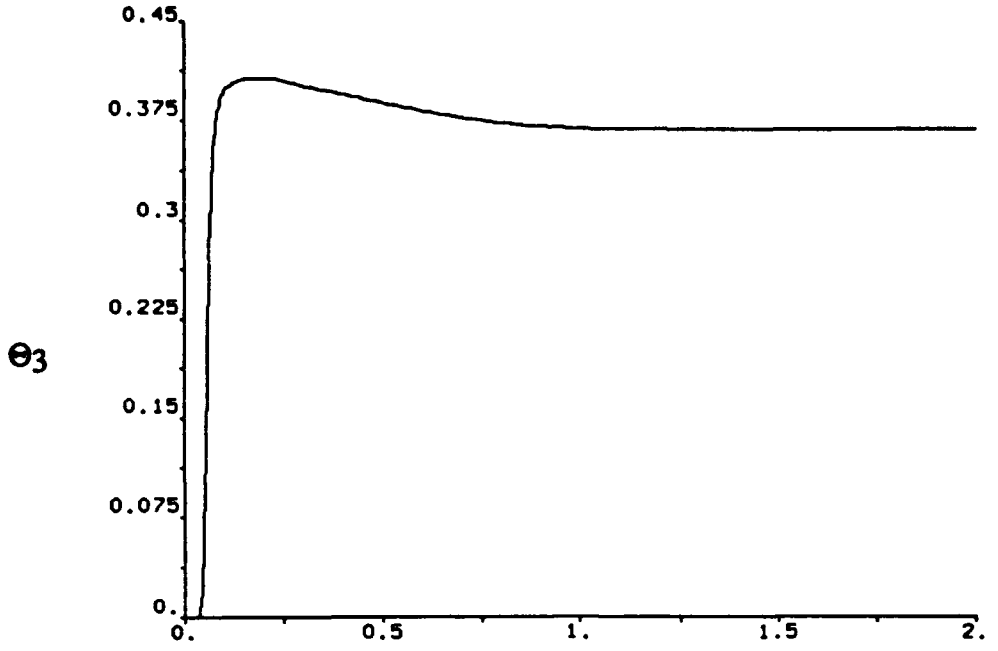


Figure 15c. Convergence of Θ_3

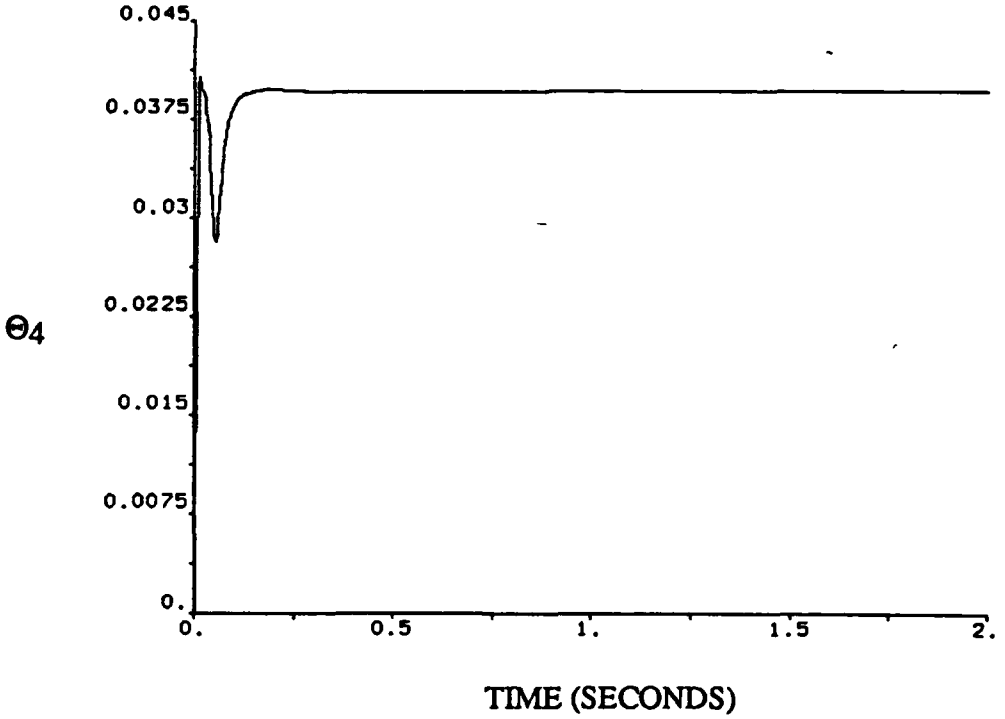


Figure 15d. Convergence of Θ_4

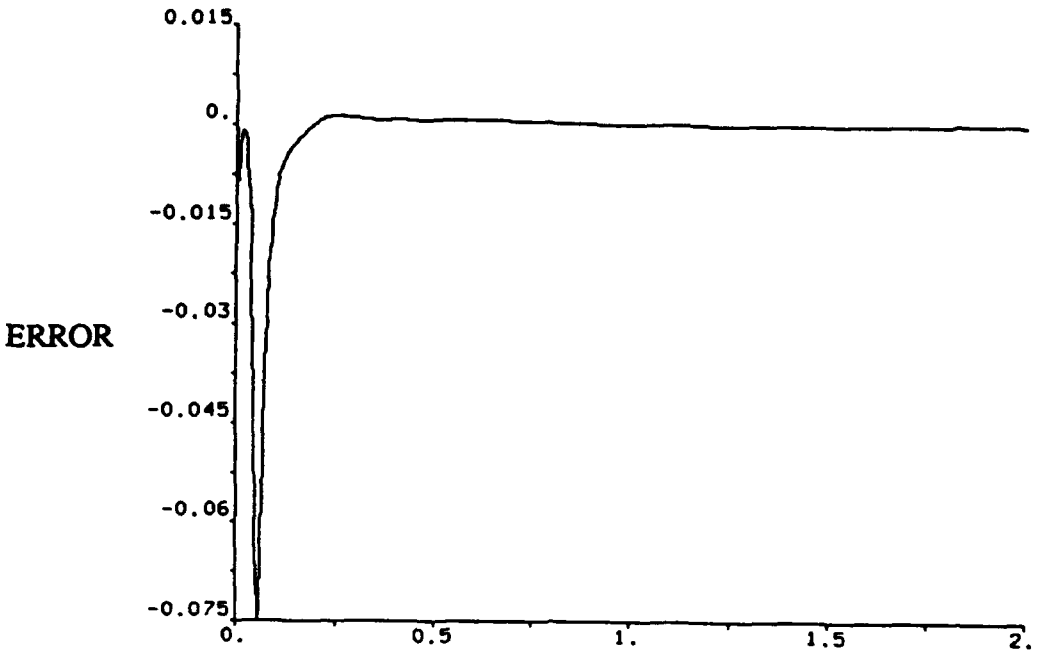


Figure 15e. Convergence of Parameter Error

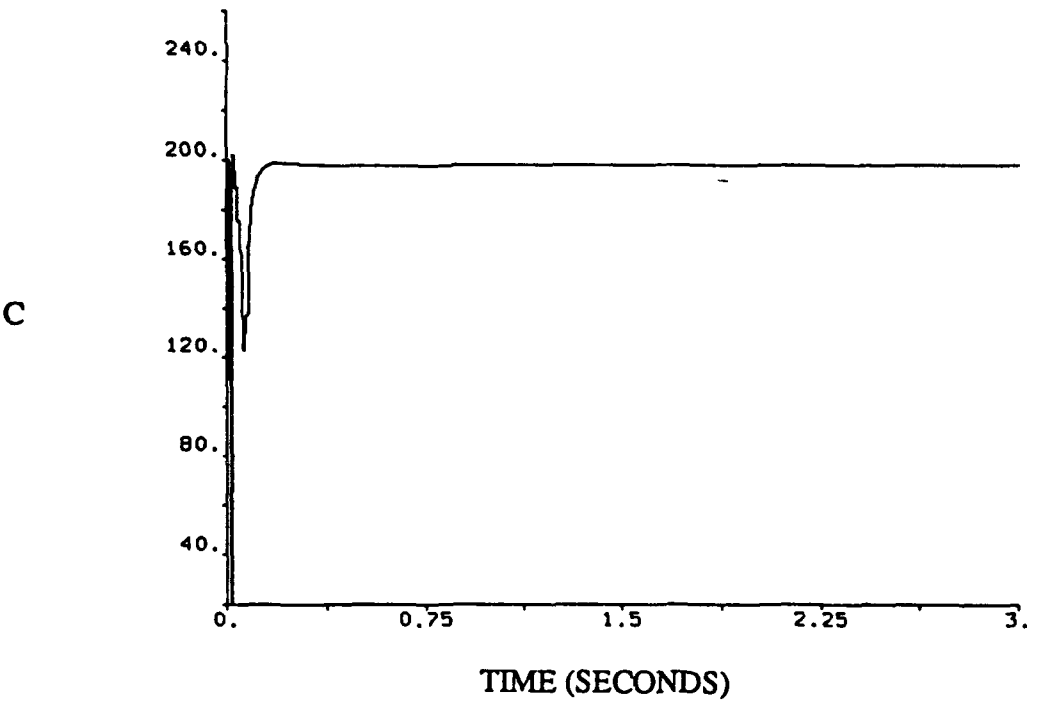


Figure 16a. Convergence of c

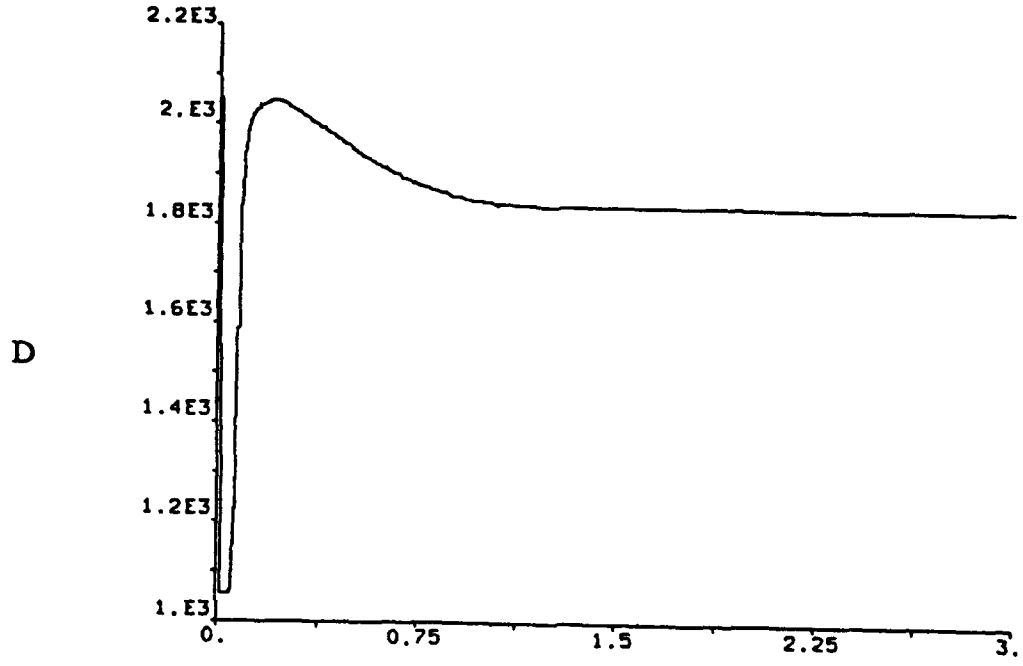


Figure 16b. Convergence of d

4. ESTIMATION OF AN UNKNOWN TIME VARYING ENVIRONMENT

The results from section 3 show that the adaptive extender architecture can interact with an unknown time invariant environment, which is accomplished through the inclusion of a least squares parameter estimator that produces values for the unknown environmental coefficients. These data are then used to update the extender controllers. Section 4 expands on this capability and addresses the behavior of an extender in the presence of an unknown time varying environment. Section 4.1 discusses the expansion of the least squares algorithm through the introduction of a forgetting factor. The resulting Bounded Gain Forgetting (BGF) estimator will be added to the baseline extender system. Simulation results from tests of this composite system are then presented, which leads to a discussion of the effects of time varying transfer functions on the extender/BGF system. An alternate approach to the time varying estimation case is also discussed, which addresses the requirements of time varying transfer functions. Simulation results are then presented to examine the behavior of the full composite system.

4.1 BOUNDED GAIN FORGETTING ESTIMATION

The standard least squares estimator used in section 3 performs poorly when tracking time varying parameters.[6] This estimation can be explained from a mathematical standpoint. As the parameter estimates settle on their values, the gain matrix P(t) converges to zero essentially shutting off the parameter convergence.[6] The least squares estimator attempts to fit all the data to the parameter estimates, from the start of convergence to the present time. Introducing a forgetting factor favors the more recent data and filters out older data allowing the estimated parameters to continue to converge to new values. The resulting P(t) matrix no longer converges to zero. Adding this forgetting factor enables the estimator to track time varying parameters. From an implementation standpoint, the forgetting factor is introduced into the gain matrix update formula. Equation (62) is rewritten yielding

$$\frac{dP}{dt} = \omega(t)\, P(t) - P(t)\, W(t)^T\, W(t)\, P(t) \qquad (90)$$

where $\omega(t)$ is the time varying forgetting factor. The parameter update formula (61) stays the same for the bounded gain case, the term $\omega(t)\, P(t)$ is added to the original gain update formula to produce the weighted result, and the selection of the weighting function $\omega(t)$ determines the behavior of the estimator. If $\omega(t)$ is set to zero, then the standard least squares estimator from

section 3 results. If the weighting function is set to a positive constant, then the resulting gain matrix goes to infinity in the absence of a persistently exciting input signal.[6]

A time varying forgetting factor is selected that filters data when the input signal is persistently exciting and remains inactive when the input is not. Based on this criteria the following form of $\omega(t)$ was chosen:

$$\omega(t) = \omega_0 \left(1 - \frac{\|P\|}{k_0} \right). \tag{91}$$

The term ω_0 is a positive constant that represents the maximum forgetting factor, k_0 represents the upper bound of the gain matrix magnitude, and $\|P\|$ represents the norm of P. For the extender implementation, the Frobenius norm was chosen. If the norm of P is small, then the forgetting speed is increased, which occurs when the input data are persistently exciting. When the norm of P becomes large and approaches k_0, then the forgetting factor is reduced. Large values of ω_0 produce faster forgetting, but allow the estimator to be influenced by noise. The constant ω_0 is chosen based on this tradeoff and k_0 has a similar effect on parameter convergence. In general, the value of k_0 can be chosen to equal to the norm of P(0). The estimator, which includes the gain update equation described by equation (90) and the forgetting factor formula described by equation (91), is known as the BGF estimator.[6]

4.2 BGF ESTIMATOR IMPLEMENTATION

The simulation implementation of the BGF estimator was similar to the least squares estimator as described in section 3. The simulation configuration used to produce the data for section 3 was a starting point for the bounded gain implementation. Formulas for the gain matrix update model PDOT from figures 13 and 14 were modified to account for the changes described by equations (90) and (91). This new gain matrix update model includes calculations for the norm of P(t), calculations for the forgetting function $\omega(t)$, and an implementation of the new gain update equation; these were the only changes required to produce the BGF system.

From a simulation standpoint, initial values for ω_0 and k_0 needed to be identified. Forms for the input signals and parameter variations also had to be selected. Several tests were performed to aid in this process. Various values for ω_0, k_0, P(0) and parameter variation were tested, and the values for these parameters were selected based on these tests.

The selection of ω_0 is based on the desired performance of the estimator. The following equation was used to select ω_0:

$$T_S \cong \frac{4}{\omega_0} \tag{92}$$

where T_S is the desired settling time of the parameter estimates. A value of 16 was chosen for ω_0, producing a settling time of 0.25 second.

Initial simulation results showed that a larger initial value for the gain matrix P(t) is required for the BGF estimator. Based on these results, an initial value of 10,000 was chosen for P(0), which produces an initial value for k_0 of 20,000 from the norm of P(0). This value is the upper limit of the gain matrix P(t). The input signal to the estimator should meet the requirements for persistent excitation. Input signal selection was based on the experiences of section 3. A sum of low-frequency sinusoids was chosen for the input signal. The input signal variations over time produce enough information data to stimulate the parameter estimator.

The final form which needed to be determined was the selection of the parameter variations. Initial tests showed that parameters vary as step or ramp functions change too quickly for the BGF estimator. The maximum parameter variation is influenced by the selection of ω_0, which controls the settling time of the estimator. If the parameter varies too quickly for a given value of ω_0, then the estimator will not be able to track these changes. Based on this performance, the parameter variation was chosen to be a low-frequency sinusoid. The following function was chosen for the parameter variation:

$$\text{Parameter Value} = \text{Base Value} + \sin\left(\frac{t}{t_0}\right) \tag{93}$$

where t_0 is a positive constant which controls the period of the parameter variations. An initial value of 2.0 was selected for t_0.

A number of simulation runs were performed for different selections of ω_0, k_0, input signal, and parameter variation to evaluate the performance of the BGF estimator. As in section 3 these tests were performed in three phases. The first phase of testing was designed to examine the performance of the BGF estimator when running alone. This configuration is shown in figure 13. The second phase of testing coupled the estimator with a simplified extender

model. Finally, the third phase of testing examined the full composite system comprising the BGF estimator and extender architecture.

In the first phase of testing, a data signal W(t) was produced from a number of input sinusoids. The parameter vector $\Theta(t)$ consisted of three positive constants and one time varying function defined by equation (93). The output signal Y(t) was produced from the product of W(t) and $\Theta(t)$. Signals W(t) and Y(t) were fed into the BGF estimator, producing estimates for the parameters $\hat{\Theta}$. BGF estimation constants were initialized to the values discussed in the beginning of this section. Figure 17 shows the performance of the BGF estimator for this setup. The BGF estimator results in figure 17 show convergence for the three constant parameters similar in behavior to the results from section 3. Figure 17 also shows that the BGF estimator tracks the time varying parameter. The plot shows that when the predicted parameter value strays from the actual value the estimator activates and converges to a new result, and this behavior produces the steps seen in the parameter track. This behavior can also be seen in a plot of the estimator error. Figure 18 is a plot of the error for this simulation run.

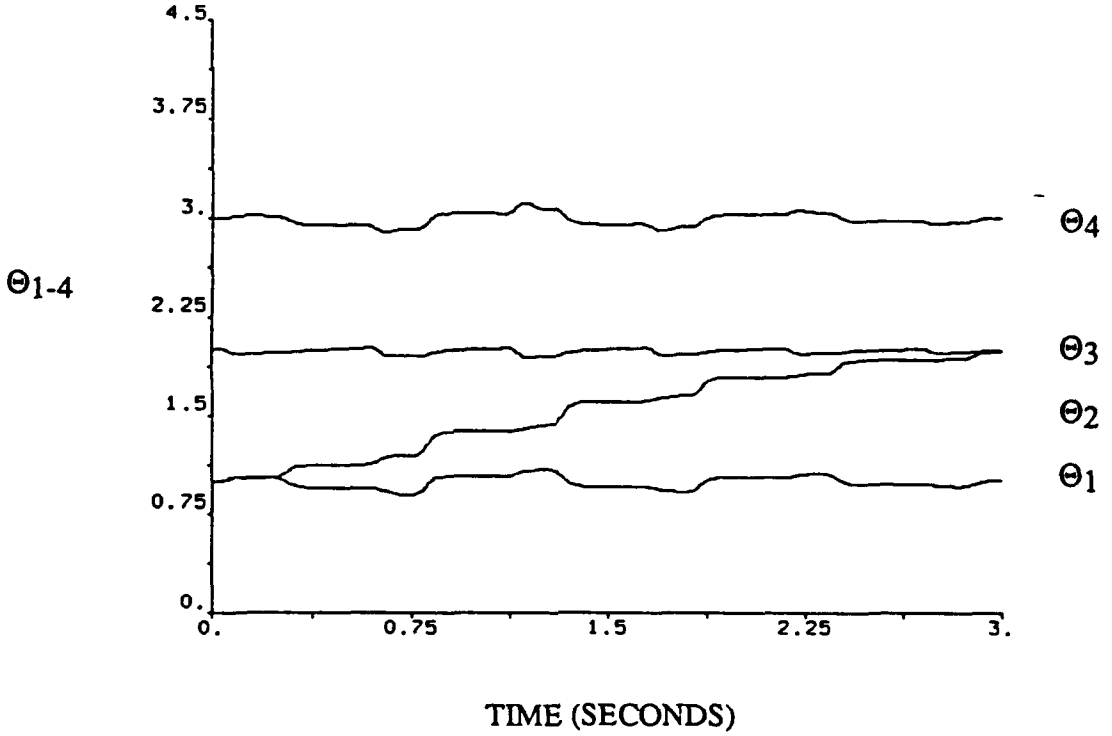


Figure 17. BGF Parameter Convergence

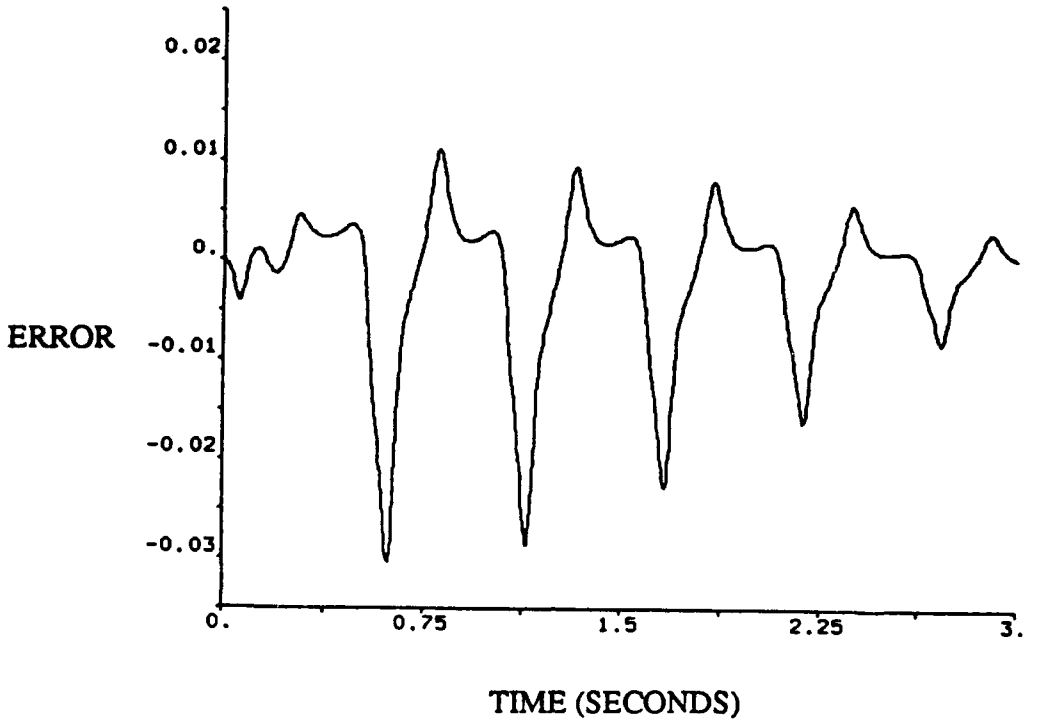


Figure 18. Parameter Convergence Error

The tracking behavior of the BGF estimator is controlled by the settings of the estimator constants. Variations in the initial gain matrix produce little effect in the resulting parameter convergence. A smaller initial gain limits the speed of convergence and results in a larger parameter error, but also reduces the amount of overshoot as the parameters converge. Conversely a larger initial gain produces faster convergence with less parameter error and more parameter overshoot.

A smaller value of ω_0 slowed the response of the estimator and resulted in a smaller step magnitude that greatly increased the parameter error and decreased the estimators ability to track the time varying parameter. Larger values of ω_0 reduced the parameter error, but also introduced oscillations in the parameter convergence. Changes in the value of ω_0 greatly affected the behavior of the BGF estimator.

Reducing the input sinusiod frequencies increased the step period of the parameter tracking and reduced the number of oscillations in the parameter error, but increased the error magnitude. Lower frequency inputs provided less information for the parameter estimator. Below a certain threshold, the input signals were not persistently exciting, which resulted in no parameter convergence. Conversely, increasing the frequency of the input sinusoids decreased

the period of the steps in the estimator track and introduced more oscillations in the parameter error, but reduced the error magnitude. Higher frequency inputs provide more information to the estimator. The upper limit on frequency was bounded by the targeted operating frequencies of the extender. The extender is designed to operate at relatively low frequencies, which places an upper limit on the information content of the input and output signals. These results show that in some cases the ideal signals for the parameter estimator and extender are mutually exclusive.

In general, these results verified the behavior of the BGF estimator implementation. The first phase of testing showed the BGF estimator's ability to track time varying parameters, and provided insight on how the BGF constants control parameter convergence. Tests were also performed with two and three parameters varying in time. Increasing the number of time varying parameters placed more demands on the estimator and required more information content from the input signals. Parameter tracking slowed as more parameters were changed from constants to time varying functions. The baseline estimator was subsequently tuned to handle the increased load, and acceptable tracking performance was achieved for these cases.

The second phase of testing coupled the BGF estimator, which was verified in the first series of tests, with the simplified extender model described by equation (68) from section 3. Parameters **c** and **d** were now produced by time varying functions described by equation (93), and the values for c(t) and d(t) were fed into state space models producing a time varying system. As in section 3, a sum of sinusoids was used as an input signal to test the simplified model. This input signal and the resulting output signal were passed to the BGF estimator, which produced the parameter estimates $\hat{\Theta}(t)$. Figure 19 shows results from a simulation test based on an initial controller model of $E_0 = 200s + 2050$. BGF estimation constants were initialized to the values discussed in the beginning of this section. The environmental parameter **c** is a constant = 200, while **d** is a time varying function = 1845 + (10 sin (t/2)). This phase of testing also included the filtering models used in section 3. Data plotted in figure 19 show that the BGF estimator performs poorly with data provided by the simplified time varying extender model. These results showed that the filtering operations were developed for time invariant models. Further tests also showed that this system exhibits similar behaviors when the BGF estimator parameters are modified. Initial estimator gain P(0), input sinusoid frequency, and estimator time constants ω_0 were varied to examine estimator performance. The results of these tests matched the behaviors described for the first phase of testing, although the estimator performance was poor. These results lead to the third set of tests performed to examine the full composite BGF/extender system.

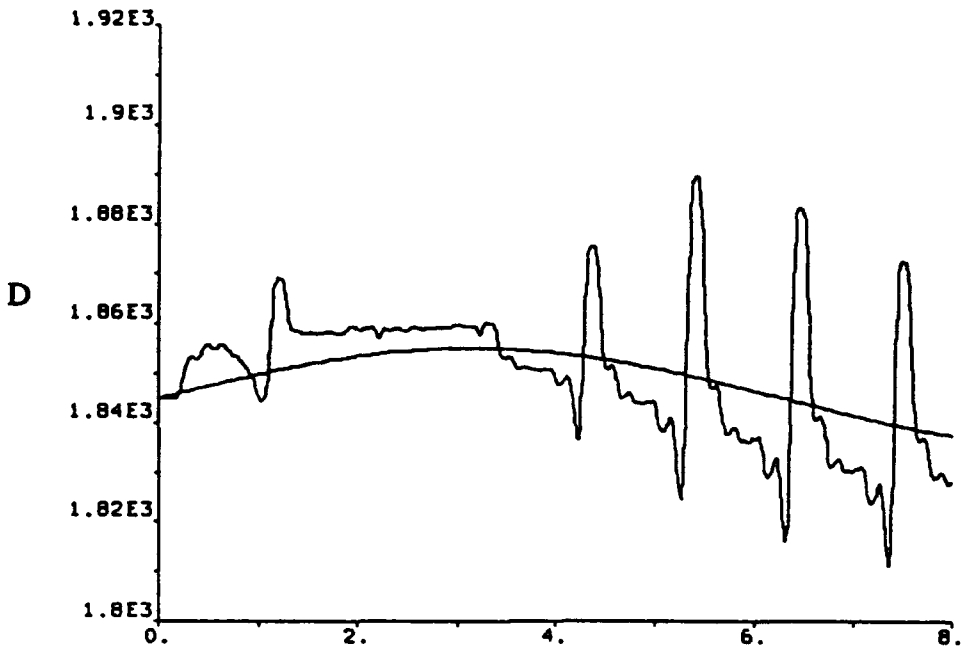


Figure 19. Simplified Time Varying Extender Parameter Results

The third and final phase of testing examined the composite extender/BGF estimator system shown in figure 14. For these tests, a time varying function was substituted for the environmental coefficients E_0 and E_1. These functions were included in the derivations from section 3.3 to produce a time varying composite system. The input signals U_h, gain matrix values P(0), and estimator time constants ω_0 and k_0 from the second phase of testing were also used for these tests. The BGF estimator implementation was the same for both the second and third phases of testing.

Data from the simulations in section 3 showed that the simplified third-order extender model described by equation (68) produced the same results as the full composite system using the same estimator implementation, which was also the case for the composite extender/BGF simulations based on the derivations from section 4. The results from the second phase of testing showed that the BGF estimator performed poorly when attempting to track the time varying coefficients **c** and **d** of the simplified third-order extender model. In the case of the composite extender/BGF system, estimated values of the time varying coefficients E_0 and E_1 did not track the actual values. The estimated coefficients showed the same behavior as the results from the second set of tests, but were offset from the correct values and showed similar poor performance.

This lack of performance prompted a fresh examination of the derivations for the time varying composite extender/estimator system.

4.3 TIME VARYING TRANSFER FUNCTION ANALYSIS

The first two phases of the initial tests verified the ability of the BGF estimator to track time varying parameters. Results from the final phase of testing showed that the formula reduction of the composite extender system described by figures 8 and 9 did not equate to the third-order transfer function described by equation (68). The transfer function reduction rules used for the time invariant case cannot be applied to the time varying case. For the time invariant case, the system block diagram from figure 8 can be reduced using standard transfer function block diagram rules, or through the application of Mason's rule to arrive at a final composite system transfer function. The transfer function described by equation (45) results. The two coefficients of the environmental model E appear in almost every term of the resulting 22nd-order transfer function. For the time invariant case, many of the poles and zeros of this transfer function cancel, resulting in the simplified third-order model. For the time varying case, this cancellation is not assured.

The block diagram in figure 8 should be reduced in a way to account for the time variation in the environment. The standard techniques mentioned previously can be applied to the time invariant portion of architecture. Figure 20 shows the resulting block diagram based on a single input F_h and a single output F_n. $E_1(t)$ is the first-order time-varying environment. H_1 and H_2 are third-order time invariant transfer functions resulting from the reduction of figure 8. $E_1(t)$, H_1, and H_2 have the following forms:

$$H_1 = \frac{-2}{Q} E_0^{-1} \frac{k_0^2}{(s + k_0)^2} = \frac{-2}{Q} \frac{k_0^2}{(as + b)(s^2 + 2k_0 s + k_0^2)} \tag{94}$$

$$H_2 = E_0^{-1} \frac{k_0^2}{(s + k_0)^2} = \frac{k_0^2}{(as + b)(s^2 + 2k_0 s + k_0^2)} \tag{95}$$

$$E_1(t) = (c(t)s + d(t)) \frac{k_1}{s + k_1} \, . \tag{96}$$

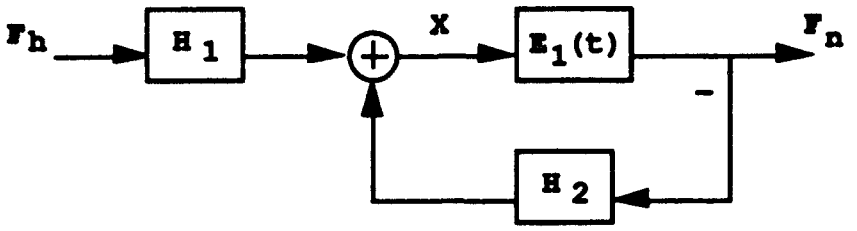


Figure 20. Reduced Time Varying Extender Architecture

These three transfer functions can then be combined using their canonical state space representations.[10] The following equations are used for this process (for the feedback connection of $E_1(t)$ and H_2):

$$A(Z) = A(E_1,H_2) = \begin{bmatrix} A_E - B_E Y_H D_H C_E & -B_E Y_H C_H \\ B_H Y_E C_E & A_H - B_H Y_E D_E C_H \end{bmatrix}$$

$$B(Z) = B(E_1,H_2) = \begin{bmatrix} B_E Y_E \\ B_H Y_E D_E \end{bmatrix}$$

$$C(Z) = C(E_1,H_2) = \begin{bmatrix} Y_E C_E & -Y_E D_E C_H \end{bmatrix}$$

$$D(Z) = D(E_1,H_2) = \begin{bmatrix} Y_E D_E \end{bmatrix} \text{ where}$$

$$Y_E = (I + D_E D_H)^{-1} \text{ and } Y_H = (I + D_H D_E)^{-1}. \quad (97)$$

Applying these equations produces a fourth-order state space description for the time varying system. This system shall be referred to as Z. The system Z is then combined in series with H_1 to produce a final composite system description using the following equations:

$$A(H_1,Z) = \begin{bmatrix} A_H & 0 \\ B_Z C_H & A_Z \end{bmatrix} \qquad B(H_1,Z) = \begin{bmatrix} B_H \\ B_Z D_H \end{bmatrix}$$

$$C(H_1,Z) = \begin{bmatrix} D_Z C_H & C_Z \end{bmatrix} \quad D(H_1,Z) = \begin{bmatrix} D_Z D_H \end{bmatrix}. \quad (98)$$

The final result of this reduction is a seventh-order state space representation of the time varying extender system. A number of the coefficients of this state space model are functions of the time varying parameters of $E_1(t)$. This state space model describes a system in the following form:

$$x(t) = A(t)\,x(t) + B(t)\,u(t), \quad (99)$$

$$y(t) = \mathbf{C}(t)\, x(t) + \mathbf{D}(t)\, u(t). \tag{100}$$

Equation (99) can be rewritten as

$$\frac{x(t) - x(t-1)}{\Delta t} = \mathbf{A}(t-1)\, x(t-1) + \mathbf{B}(t-1)\, u(t-1). \tag{101}$$

Solving equation (101) to x(t) and substituting the result into equation (100) yields

$$y(t) = \mathbf{C}(t)\left[\Delta t(\mathbf{A}(t-1)x(t-1) + \mathbf{B}(t-1)u(t-1)) + x(t-1)\right] + \mathbf{D}(t)\, u(t). \tag{102}$$

If Dt is small enough, then the values of the state space model coefficients at time t are close to their values at time t - Δt. Based on this assumption equation (102) can be rewritten as

$$y(t) \cong \mathbf{C}(t-1)\left[\Delta t(\mathbf{A}(t-1)\, x(t-1) + \mathbf{B}(t-1)\, u(t-1)) + x(t-1)\right] + \mathbf{D}(t-1)\, u(t-1). \tag{103}$$

Multiply the terms of equation (103) and rearrange the resulting equation to produce an equation of the following form:

$$Y(t) \cong W(t)\, \Theta(t) \tag{104}$$

where W(t) consists of functions of the input u(t) and the state variables x(t). The variable $\Theta(t)$ from equation (104) is a vector of the unknown parameters that map back to the unknown coefficients of the environment $E_1(t)$.

This approach to deriving a system form that can be used for estimation produces a result of the proper form; however, a problem is also introduced which makes it unfeasible. The measurement vector W(t) is a function of the state variables of the composite system that cannot be extracted directly from the extender system. A state observer could be used to produce values for these state variables, but the results would be functions of the unknown coefficients of $E_1(t)$. This mutual dependence of state variables and unknown parameters introduces the problem of having to estimate both sets of unknowns at the same time, and this dependence in itself is a complex problem that goes beyond the scope of this report. The examination of time varying model and state space descriptions does lead to an alternative solution for the extender system.

4.4 UTILIZATION OF MULTI-INPUT DATA

As part of the investigation of time varying state space models, the extender system was examined to determine if the unknown state variables could be observed. Two signals that are present and measurable in the extender experimental architecture can provide the data required to estimate the unknown time varying coefficients of E_1. Figure 6 shows that both the extender position and velocity are fed back to provide primary compensation for the extender system. Figure 7 shows that these signals are fed forward through the environmental coefficients to produce the output F_n. Figure 21 shows this relation.

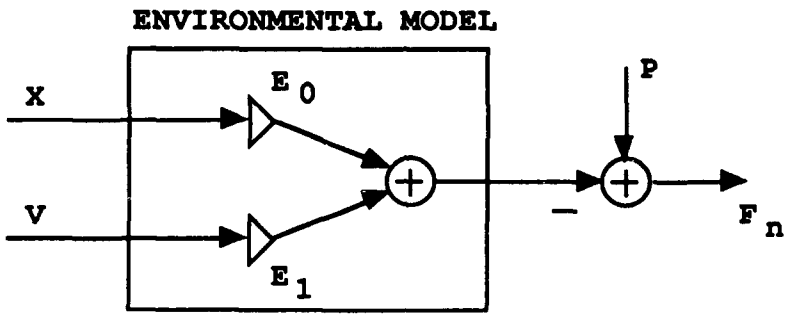


Figure 21. Extender Environmental Relation

The output F_n can be written as a function of the position X and velocity V in the following form:

$$F_n = P - (E_0 X) - (E_1 V). \tag{105}$$

Equation (105) can easily be rewritten in the desired estimator form:

$$Y = F_n = W \Theta = [\, X \quad V \,] \begin{bmatrix} E_0 \\ E_1 \end{bmatrix}. \tag{106}$$

This simple second-order model can be implemented as a BGF estimator described in section 4.1. The estimator utilizes the extender position and velocity to produce estimates of the unknown time varying coefficients E_0 and E_1. Previous implementations of the extender/ estimator system were of a higher order, but the input signals which were part of W(t) were created from filtered values of a single input F_h. This approach uses two separate unfiltered input signals X and V to form W(t).

This approach was implemented in Simnon and tested using the same configuration that was used in the initial tests of the BGF/extender system. These tests uncovered another requirement that is placed on the extender system for parameter convergence when position and velocity are used as inputs. The first simulation runs used a sum of unit magnitude sinusoids for an input signal. In past tests, the force amplification produced an output signal with similar orders of magnitude. Variations in the input and output forces were on the order of 10^1. The resulting position and velocity signals used as inputs to the multi-input estimator system varied on the order of 10^{-3}, but these input signals did not provide enough information for persistent excitation. This problem was solved by increasing the input magnitude of the input sinusoids by a factor of 10, which caused an associated increase in the magnitude of the position and velocity signals. At these levels, sufficient information was provided to allow for parameter convergence and tracking.

Simulation runs were performed to test the multi-input approach with the new input signals. Figure 22 shows the results of the multi-input BGF/extender simulation. The parameter E_0 was replaced with the same time varying function that was used for the initial tests: $E_0 = 1845 + (10.0 * \sin(t/2))$. The frequencies of the input signals and the selections of $\omega_0 = 16$ and $k_0 = 12{,}000$ remained the same as in past tests. Figures 22a and 22b show that accurate estimates for time varying parameters can be generated for the extender system based on the extender position and velocity, which places an additional requirement on the magnitude of the input F_h. Small variations in the input force may not provide sufficient information for parameter convergence based on position and velocity signals. This technique represents an alternative that addresses the requirements of the time varying parameter case, places the same conditions on the input signal F_h for the time invariant case, and represents a potential tradeoff in capability. This approach can be applied to the time invariant case described in section 3.

Updating the controllers with the parameter estimates allows the extender to adapt to an unknown time varying environment. The BGF estimator tracks the changes in the environment and generates parameter values. These estimates allow the extender controllers to maintain the proportional force relation in continuous time. Using the extender velocity and position as inputs provides sufficient data to allow the extender system to track environmental changes.

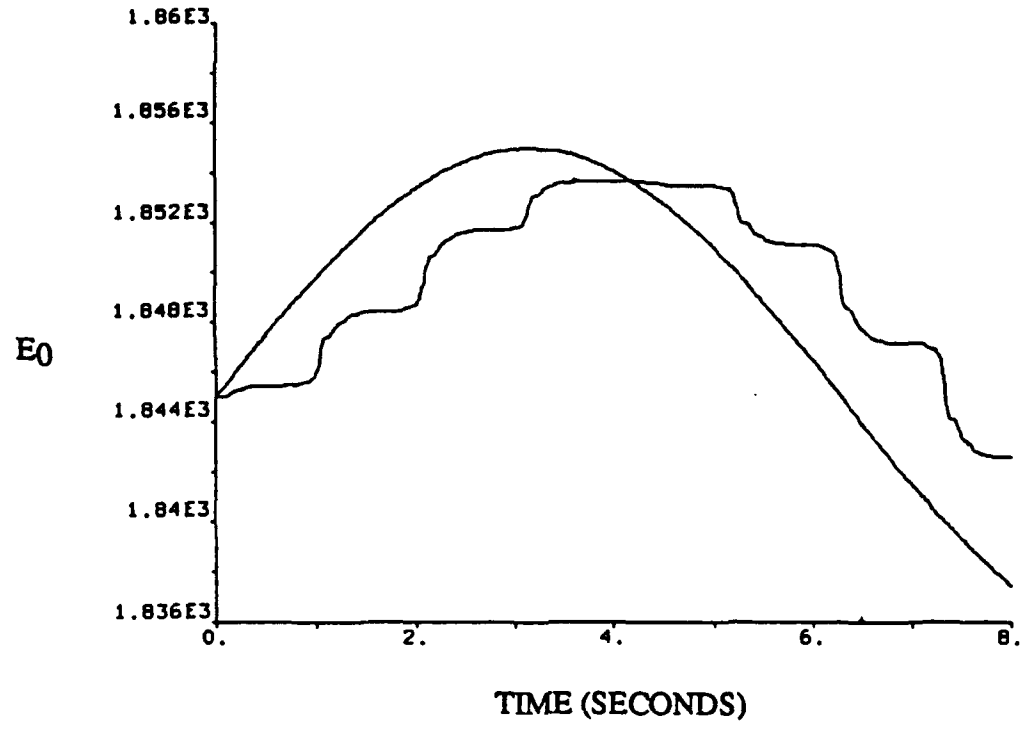


Figure 22a. Parameter Tracking of E_0

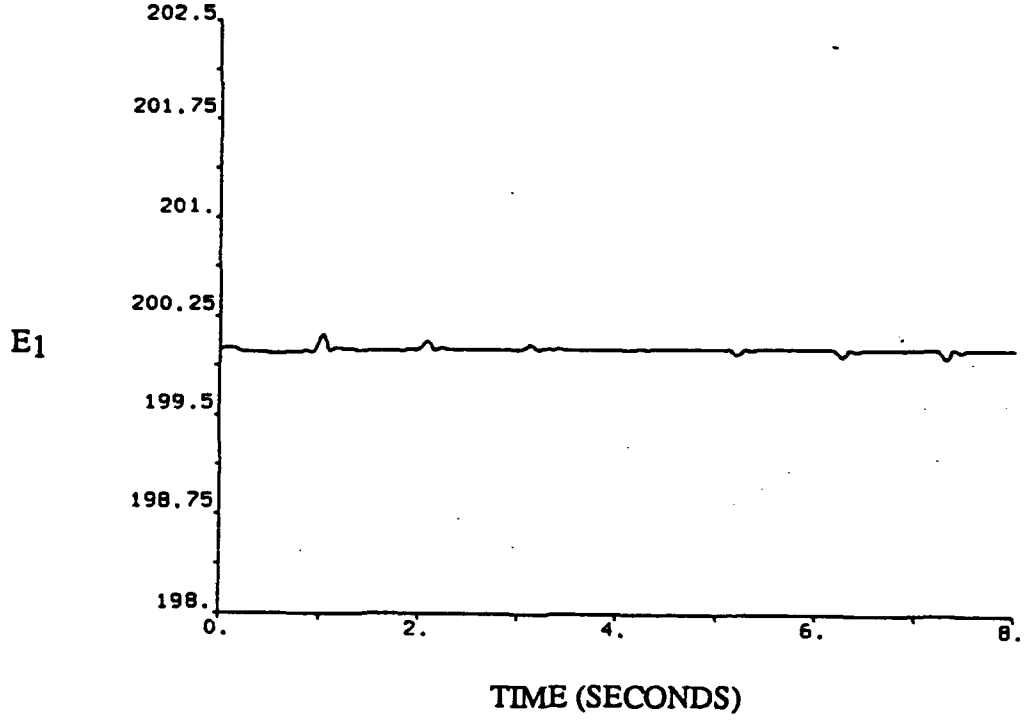


Figure 22b. Parameter Tracking of E_1

5. CONCLUSION

The extender architecture developed by Kazerooni describes a telerobotic system that produces force amplification by using a single set of actuators, which improves upon the traditional master/slave architectures. The extender allows the human operator to perform tasks beyond the physical capabilities of the human body. In this document, it is reported that the extender architecture can be modified to produce a force to position relationship, allowing the input force of the operator to map to the motion of the extender in the absence of an external contact force. Presently, these two configurations are mutually exclusive. Each of these capabilities can be achieved with either digital or continuous controllers. Reliance of the extender system of a complete model of its operating environment places a limitation on its usefulness. This report has approached this problem in two steps.

5.1 SUMMARY

When dealing with an unknown time invariant environment, a least squares estimator can be used to produce estimates of the environmental parameters. The input and output forces provide information for the estimator. When the estimator has converged to a solution based on parameter error, the extender controllers are updated. The new controllers maintain the force relationship based on the parameter estimates. If the information signal is persistently exciting, then parameter convergence can be achieved.

In the case of an unknown time varying environment, a similar approach was examined. A forgetting factor was added to the least squares estimator producing a BGF estimator, which displayed the ability to track time varying parameters in tests using a simplified extender model. When coupled with the composite extender system, the BGF estimator tests lead to the realization that the transfer function reduction of time varying composite systems did not fit into the framework used for the time invariant case. Time varying state variables used to produce the parameter estimates also depended on the values of these estimates. The simultaneous estimation of unknown model parameters and state space variables introduced a level of complexity that is beyond the scope of this report. An alternate choice of input signals consisting of extender position and velocity avoids the time varying transfer function problem. Using these inputs, a BGF estimator can track time varying environmental model parameters. Values of the parameter estimates are then used to update the extender controllers to maintain the force relationship in continuous time. As with the time invariant case, characteristics of the input signal determine the level of parameter convergence. The selection of position and velocity as input signals

introduces additional constraints on the input signal for parameter convergence.

In general, the extender system can be improved with the application of parameter estimation. The estimator can produce values for unknown environmental model parameters. These data when applied to the extender controllers allow the extender to adapt to an unknown time varying environment. The adaptive extender system could then operate in differing environments without changing its performance characteristics.

5.2 FURTHER DEVELOPMENT

This report represents an initial investigation into the development of an adaptive system based on Kazerooni's extender architecture, and the results are closely tied to the experimental framework used to verify the extender system. There are a number of areas where new technology could be applied to improve upon the baseline extender idea resulting in a more capable system. These areas include the following:

1. The application of joint torque sensing could replace external force sensing, which would eliminate the reliance on measuring force at the point of contact with the environment. This procedure removes the limitation on how or where the extender interacts with the environment.

2. New types of actuators based on harmonic drives or possibly shape memory alloy could greatly reduce the size of the hardware used to produce extender motion. Currently, the extender system is anchored to a fixed base. The inclusion of these actuators may allow for greater mobility. The extender controller design is closely tied to the hardware configuration. New hardware designs could also introduce new areas of investigation including a multi-input multi-output extender system.

3. The results presented in this report show that adaptive control techniques can be applied to the extender system, but these results do not represent a final or completely robust solution. Force amplification and force/position relations should be combined into a composite system. Estimator settings such as gain could be updated through the application of neural nets, which would tune parameter convergence based on characteristics of parameter error. This optimization of estimator settings for parameter convergence would be based on learned experience of extender operation. The problem encountered with the single input time varying approach based on extender input force should be addressed. These improvements would result in a robust controller implementation that could operate in a truly dynamic environment.

The results of the these adaptations would be a self sufficient exoskeletal suit. This telerobotic device would be mobile and would enable its operator to perform a wide variety of tasks in differing environments providing the ability to place human experience into areas where traditional approaches removed man from the loop and replaced him with a machine. This technology could have a number of applications in areas such as space exploration and development. The system results in the combination of the best features of both, the intellect of man coupled with the physical attributes of the machine.

REFERENCES

[1] H. Kazerooni, "Stability and Performance of Robotic Systems Worn by Humans," *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 558-563, 1990.

[2] H. Kazerooni, "Human/Robot Interaction via the Transfer of Power and Information Signals Part I: Dynamics and Control Analysis," *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 1632-1640, 1989.

[3] H. Kazerooni, "Human/Robot Interaction via the Transfer of Power and Information Signals Part II: An Experimental Analysis," *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 1641-1647, 1989.

[4] H. Kazerooni and W. K. Foslienm, "On the Control and Stability of Robots Worn by Humans: Theory," *American Control Conference*, vol. 3, pp. 1918-1924, 1989.

[5] H. Kazerooni and T. M. Hessburg, "On the Control and Stability of Robots Worn by Humans: Experiments," *American Control Conference*, vol. 3, pp. 1925-1930, 1989.

[6] J. E. Slotine and W. Li, *Applied Nonlinear Control,* Prentice-Hall, Englewood Cliffs, NJ, 1991.

[7] L. Jackson, *Digital Filters and Signal Processing*, Kluwer Academic Publishers, Norwell, MA, 1989.

[8] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *"Feedback Control of Dynamic Systems,"* Addison-Wesley, Reading, MA, 1986.

[9] L. Ljung and T. Soderstrom, *"Theory and Practice of Recursive Identification,"* MIT Press, Cambridge, MA, 1983.

[10] C. T. Chen, *"Linear System Theory and Design,"* Holt, Rinehart and Winston, Orlando, FL, 1984.

BIBLIOGRAPHY

Astrom, K. J., and B. Wittenmark, *Adaptive Control*, Addison-Wesley, Reading, MA, 1989.

Chen, C. T., *Linear System Theory and Design*, Holt, Rinehart and Winston, Orlando, FL, 1984.

Franklin, G. F., J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, Addison-Wesley, Reading, MA, 1986.

Jackson, L., *Digital Filters and Signal Processing*, Kluwer Academic Publishers, Norwell MA, 1989.

Jansen, J. F., and J. N. Herndon, "Design of a Telerobotic Controller with Joint Torque Sensors," *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 1109-1115, 1990.

Kazerooni, H., "Human/Robot Interaction via the Transfer of Power and Information Signals Part I: Dynamics and Control Analysis," *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 1632-1640, 1989.

Kazerooni, H., "Human/Robot Interaction via the Transfer of Power and Information Signals Part II: An Experimental Analysis," *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 1641-1647, 1989.

Kazerooni, H., "Stability and Performance of Robotic Systems Worn by Humans," *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 558-563, 1990.

Kazerooni, H., T. B. Sheridan, and P. K. Houpt, "Robust Compliant Motion for Manipulators, Part I The Fundamental Concepts of Compliant Motion," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 2, pp. 83-92, June 1986.

Kazerooni, H., T. B. Sheridan, and P. K. Houpt, "Robust Compliant Motion for Manipulators, Part II Design Method," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 2, pp. 93-105, June 1986.

Kazerooni, H., and T. M. Hessburg, "On the Control and Stability of Robots Worn by Humans: Experiments," *American Control Conference*, vol. 3, pp. 1925-1930, 1989.

Kazerooni, H., and W. K. Foslienm, "On the Control and Stability of Robots Worn by Humans: Theory," *American Control Conference*, vol. 3, pp. 1918-1924, 1989.

Ljung, L., *System Identification Theory for the User*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

Ljung, L., and T. Soderstrom, *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, MA, 1983.

Narendra, K. S., and A. M. Annaswamy, *Stable Adaptive Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

Oppenheim, A., and A. Willsky, *Signal and Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

Repperger, D. W., S. J. Remis, and G. Merril, "Performance Measures of Teleoperation Using an Exoskeleton Device," *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 552-557, 1990.

Slotine, J. E., and W. Li, *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, NJ, 1991.

Reverse Blank

APPENDIX A
SIMNON SIMULATION CODE

This appendix contains the Simnon source code implementations of the model described in this report. These listings appear in the same order as the material presented in the text of this paper. References to figure numbers are included with each listing.

Second-Order Human Arm Dynamics Model "T" (figure 3)

```
Continuous System TK2
input u
output y
state x1 x2
der x1d x2d
c1=k*k*(0.572-0.04089*2*k)
c2=k*k*(2-0.04089*k*k)
a1=-1*2*k
a2=-1*k*k
d=k*k*0.04089
y=(c1 * x1) + (c2 * x2) + (d * u)
x1d=(a1 * x1) + (a2 * x2) + u
x2d=(a3 * x1) + (a4 * x2)
a3:1.0
a4:0.0
k:100.0
end
```

Third-Order Force-Force Extender Controller "H_h" (figure 5)

```
Continuous System HH
input u
output y
state x1 x2 x3
der x1d x2d x3d
d=0.00000026511*k*k
c1=k*k*(0.0000094247-0.00000026511*(2*k+10.25))
c2=k*k*(0.00042265-0.00000026511*(k*k+20.5*k))
c3=k*k*(0.005-0.00000026511*(k*k*10.25))
a1=-1*(2*k+10.25)
a2=-1*(k*k+20.5*k)
a3=-1*k*k*10.25
y=(2/z*c1*x1) + (2/z*c2*x2) + (2/z*c3*x3) + (2/z*d*u)
x1d=(a1 * x1) + (a2 * x2) + (a3 * x3) + u
x2d=(a4 * x1) + (a5 * x2) + (a6 * x3)
x3d=(a7 * x1) + (a8 * x2) + (a9 * x3)
a4:1.0
a5:0.0
a6:0.0
a7:0.0
a8:1.0
```

```
a9:0.0
z:0.5
k:100.0
end
```

Third-Order Force-Force, Force-Position Extender Controller "H_n" (figure 5)

```
Continuous System HN
input u
output y
state x1 x2 x3
der x1d x2d x3d
d=0.00000026511*k*k
c1=k*k*(0.0000094247-0.00000026511*(2*k+10.25))
c2=k*k*(0.00042265-0.00000026511*(k*k+20.5*k))
c3=k*k*(0.005-0.00000026511*(k*k*10.25))
a1=-1*(2*k+10.25)
a2=-1*(k*k+20.5*k)
a3=-1*k*k*10.25
y=(c1 * x1) + (c2 * x2) + (c3 * x3) + (d * u)
x1d=(a1 * x1) + (a2 * x2) + (a3 * x3) + u
x2d=(a4 * x1) + (a5 * x2) + (a6 * x3)
x3d=(a7 * x1) + (a8 * x2) + (a9 * x3)
a4:1.0
a5:0.0
a6:0.0
a7:0.0
a8:1.0
a9:0.0
k:100.0
end
```

Third-Order Force-Position Extender Controller "H_h" (figure 5)

```
Continuous System HH
input u
output y
state x1 x2 x3
der x1d x2d x3d
d=0.000053*k*k
c1=k*k*(0.0018-0.000053*(2*k+10.25))
c2=k*k*(0.0845-0.000053*(k*k+20.5*k))
c3=k*k*(1.0-0.000053*(k*k*10.25))
a1=-1*3*k
a2=-1*3*k*k
a3=-1*k*k*k
y=(2/z*c1*x1) + (2/z*c2*x2) + (2/z*c3*x3) + (2/z*d*u)
x1d=(a1 * x1) + (a2 * x2) + (a3 * x3) + u
x2d=(a4 * x1) + (a5 * x2) + (a6 * x3)
x3d=(a7 * x1) + (a8 * x2) + (a9 * x3)
a4:1.0
```

```
a5:0.0
a6:0.0
a7:0.0
a8:1.0
a9:0.0
z:0.5
k:10.0
end
```

Third-Order Discrete Force-Force Extender Controller "H_h" (figure 5)

```
Discrete System HH
input u
output y
state x1 x2 x3
new x1d x2d x3d
time t
tsamp ts
k1=8*a+4*b*tk+2*c*tk*tk+d*tk*tk*tk
k2=-24*a-4*b*tk+2*c*tk*tk+3*d*tk*tk*tk
k3=24*a-4*b*tk-2*c*tk*tk+3*d*tk*tk*tk
k4=-8*a+4*b*tk-2*c*tk*tk+d*tk*tk*tk
k5=-24-4*e*tk+2*f*tk*tk+3*g*tk*tk*tk
k6=24-4*e*tk-2*f*tk*tk+3*g*tk*tk*tk
k7=-8+4*e*tk-2*f*tk*tk+g*tk*tk*tk
k8=8+4*e*tk+2*f*tk*tk+g*tk*tk*tk
e=2*k+10.25
f=k*k+20.5*k
g=k*k*10.25
ts=t+dt
dd=k1/k8*k*k
c1=k*k*(k2/k8-k1/k8*k5/k8)
c2=k*k*(k3/k8-k1/k8*k6/k8)
c3=k*k*(k4/k8-k1/k8*k7/k8)
a1=-1*(k5/k8)
a2=-1*(k6/k8)
a3=-1*(k7/k8)
y=(2/z*c1*x1) + (2/z*c2*x2) + (2/z*c3*x3) + (2/z*d*u)
x1d=(a1 * x1) + (a2 * x2) + (a3 * x3) + u
x2d=(a4 * x1) + (a5 * x2) + (a6 * x3)
x3d=(a7 * x1) + (a8 * x2) + (a9 * x3)
dt:0.01
tk:0.01
a:0.0000002651
b:0.000009425
c:0.0004226
d:0.005
a4:1.0
a5:0.0
a6:0.0
a7:0.0
a8:1.0
```

```
a9:0.0
k:100.0
z:0.5
end
```

Third-Order Discrete Force-Force Extender Controller "H_n" (figure 5)

```
Discrete System HN
input u
output y
state x1 x2 x3
new x1d x2d x3d
time t
tsamp ts
k1=8*a+4*b*tk+2*c*tk*tk+d*tk*tk*tk
k2=-24*a-4*b*tk+2*c*tk*tk+3*d*tk*tk*tk
k3=24*a-4*b*tk-2*c*tk*tk+3*d*tk*tk*tk
k4=-8*a+4*b*tk-2*c*tk*tk+d*tk*tk*tk
k5=-24-4*e*tk+2*f*tk*tk+3*g*tk*tk*tk
k6=24-4*e*tk-2*f*tk*tk+3*g*tk*tk*tk
k7=-8+4*e*tk-2*f*tk*tk+g*tk*tk*tk
k8=8+4*e*tk+2*f*tk*tk+g*tk*tk*tk
e=2*k+10.25
f=k*k+20.5*k
g=k*k*10.25
ts=t+dt
dd=k1/k8*k*k
c1=k*k*(k2/k8-k1/k8*k5/k8)
c2=k*k*(k3/k8-k1/k8*k6/k8)
c3=k*k*(k4/k8-k1/k8*k7/k8)
a1=-1*(k5/k8)
a2=-1*(k6/k8)
a3=-1*(k7/k8)
y=(c1 * x1) + (c2 * x2) + (c3 * x3) + (d * u)
x1d=(a1 * x1) + (a2 * x2) + (a3 * x3) + u
x2d=(a4 * x1) + (a5 * x2) + (a6 * x3)
x3d=(a7 * x1) + (a8 * x2) + (a9 * x3)
dt:0.01
tk:0.01
a:0.0000002651
b:0.000009425
c:0.0004226
d:0.005
a4:1.0
a5:0.0
a6:0.0
a7:0.0
a8:1.0
a9:0.0
k:100.0
end
```

Second-Order Experimental Extender Dynamics Model "G_{dh}" (figure 6)

```
Continuous System GDH
input u
output y
state x1 x2
der x1d x2d
y=(c1 * x1) + (c2 * x2)
x1d=(a1 * x1) + (a2 * x2) + u
x2d=(a3 * x1) + (a4 * x2)
c1:0.0008925
c2:0.02106
a1:-35.549
a2:-1560.25
a3:1.0
a4:0.0
end
```

Second-Order Experimental Extender Dynamics Model "G_{dn}" (figure 6)

```
Continuous System GDN
input u
output y
state x1 x2
der x1d x2d
y=(c1 * x1) + (c2 * x2)
x1d=(a1 * x1) + (a2 * x2) + u
x2d=(a3 * x1) + (a4 * x2)
c1:0.0026775
c2:0.06319
a1:-35.549
a2:-1560.25
a3:1.0
a4:0.0
end
```

Second-Order Experimental Extender Dynamics Model "G_p" (figure 6)

```
Continuous System GP
input u
output y
state x1 x2
der x1d x2d
y=(c1 * x1) + (c2 * x2)
x1d=(a1 * x1) + (a2 * x2) + u
x2d=(a3 * x1) + (a4 * x2)
c1:0.0
c2:553888.75
a1:-35.549
a2:-1560.25
```

```
a3:1.0
a4:0.0
end
```

First-Order Integrator Model (figure 6)

```
Continuous System SS
input u
output y
state x1
der x1d
y=x1
x1d=u
end
```

Experimental Composite System Connections (figure 9)

```
Connecting system CON
time t
out=y[ss]
uh=y[sins]
sum=y[gdh]+y[gdn]+y[gp]
u[gp]=kb*kda*(k1*(k0*(y[hh]+y[hn])=kio*y[ss])-k2*kad*kt*sum)
u[gdh]=uh-y[tk2]
u[hh]=uh=y[tk2]
u[gdn]=p-(e0*y[ss]+e1*sum)
u[hn]=p-(e0*y[ss]+e1*sum)
u[tk2]=y[ss]
u[ss]=sum
kb:0.00465
kda:0.004883
k1:0.94
k0:1592.0
kio:1592.0
k2:0.00977
kad:1638.4
kt:0.169
e0:2050.0
e1:200.0
p:0.0
end
```

Input Sinusoid Generator for Input Signal uh
(magnitudes, frequencies, and phases can be altered to suit a given simulation run)

```
Continuous System SINS
output sqr qq xx yy zz vv ww x y z
time t
vv = i*sin(n*(t-phi))
ww = h*sin(o*(t-phe))
qq = g*sin(p*(t-phi))
```

```
sqr = a*sin(t*(t-phum))
x = b*sin(u*(t-phe))
y = a*sin(w*(t-phi))
z = c*sin(v*(t-pho))
xx = e*sin(q*(t-phe))
yy = d*sin(r*(t-phi))
zz = f*sin(s*(t-pho))
a:1
b:1
c:1
d:1
e:1
f:1
g:1
h:1
i:1
n:9.0
o:8.0
p:7.0
q:4.0
r:5.0
s:6.0
u:1.0
v:3.0
w:2.0
phe:0
phi:0
pho:0
phum:0
end
```

Fourth-Order Gain Update Model "PDOT" (figure 13)

```
Continuous System PDOT
input yf1 yf2 uf1 uf2
output pd1 pd2 pd3 pd4 pd5 pd6 pd7 pd8 pd9 pd10 pd11 pd12
output pd13 pd14 pd15 pd16
state p1 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13 p14 p15 p16
der p1d p2d p3d p4d p5d p6d p7d p8d p9d p10d p11d p12d p13d
der p14d p15d p16d
pd1=p1
pd2=p2
pd3=p3
pd4=p4
pd5=p5
pd6=p6
pd7=p7
pd8=p8
pd9=p9
pd10=p10
pd11=p11
pd12=p12
pd13=p13
```

```
pd14=p14
pd15=p15
pd16=p16
w1=yf1*yf1
w2=yf2*yf1
w3=uf1*yf1
w4=uf2*yf1
w5=yf2*yf2
w6=uf1*yf2
w7=uf2*yf2
w8=uf1*uf1
w9=uf2*uf1
w10=uf2*uf2
i1=(p1 * w1) + (p5 * w2) + (p9 * w3) + p13 * w4)
i2=(p2 * w1) + (p6 * w2) + (p10 * w3) + p14 * w4)
i3=(p3 * w1) + (p7 * w2) + (p11 * w3) + p15 * w4)
i4=(p4 * w1) + (p8 * w2) + (p12 * w3) + p16 * w4)
i5=(p1 * w2) + (p5 * w5) + (p9 * w6) + p13 * w7)
i6=(p2 * w2) + (p6 * w5) + (p10 * w6) + p14 * w7)
i7=(p3 * w2) + (p7 * w5) + (p11 * w6) + p15 * w7)
i8=(p4 * w2) + (p8 * w5) + (p12 * w6) + p16 * w7)
i9=(p1 * w3) + (p5 * w6) + (p9 * w8) + p13 * w9)
i10=(p2 * w3) + (p6 * w6) + (p10 * w8) + p14 * w9)
i11=(p3 * w3) + (p7 * w6) + (p11 * w8) + p15 * w9)
i12=(p4 * w3) + (p8 * w6) + (p12 * w8) + p16 * w9)
i13=(p1 * w4) + (p5 * w7) + (p9 * w9) + p13 * w10)
i14=(p2 * w4) + (p6 * w7) + (p10 * w9) + p14 * w10)
i15=(p3 * w4) + (p7 * w7) + (p11 * w9) + p15 * w10)
i16=(p4 * w4) + (p8 * w7) + (p12 * w9) + p16 * w10)
p1d=-1*((i1 * p1) + (i5 * p2) + (i9 * p3) + (i13 * p4))
p2d=-1*((i2 * p1) + (i6 * p2) + (i10 * p3) + (i14 * p4))
p3d=-1*((i3 * p1) + (i7 * p2) + (i11 * p3) + (i15 * p4))
p4d=-1*((i4 * p1) + (i8 * p2) + (i12 * p3) + (i16 * p4))
p5d=-1*((i1 * p5) + (i5 * p6) + (i9 * p7) + (i13 * p8))
p6d=-1*((i2 * p5) + (i6 * p6) + (i10 * p7) + (i14 * p8))
p7d=-1*((i3 * p5) + (i7 * p6) + (i11 * p7) + (i15 * p8))
p8d=-1*((i4 * p5) + (i8 * p6) + (i12 * p7) + (i16 * p8))
p9d=-1*((i1 * p9) + (i5 * p10) + (i9 * p11) + (i13 * p12))
p10d=-1*((i2 * p9) + (i6 * p10) + (i10 * p11) + (i14 * p12))
p11d=-1*((i3 * p9) + (i7 * p10) + (i11 * p11) + (i15 * p12))
p12d=-1*((i4 * p9) + (i8 * p10) + (i12 * p11) + (i16 * p12))
p13d=-1*((i1 * p13) + (i5 * p14) + (i9 * p15) + (i13 * p16))
p14d=-1*((i2 * p13) + (i6 * p14) + (i10 * p15) + (i14 * p16))
p15d=-1*((i3 * p13) + (i7 * p14) + (i11 * p15) + (i15 * p16))
p16d=-1*((i4 * p13) + (i8 * p14) + (i12 * p15) + (i16 * p16))
end
```

Fourth-Order Parameter Update Model "AHAT" (figure 13)

```
Continuous System AHAT
input yf1 yf2 uf1 uf2 y yh p1 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11
input p12 p13 p14 p15 p16
output ah1 ah2 ah3 ah4
```

```
state a1 a2 a3 a4
der a1d a2d a3d a4d
e1=yh-y
ah1=a1
ah2=a2
ah3=a3
ah4=a4
a1d=-1*e1*((yf1 * p1)+(yf2 * p2)+(uf1 * p3)+(uf2 * p4))
a2d=-1*e1*((yf1 * p5)+(yf2 * p6)+(uf1 * p7)+(uf2 * p8))
a3d=-1*e1*((yf1 * p9)+(yf2 * p10)+(uf1 * p11)+(uf2 * p12))
a4d=-1*e1*((yf1 * p13)+(yf2 * p14)+(uf1 * p15)+(uf2 * p16))
end
```

Parameter Error Model "YHAT" (figure 13)

```
Continuous System YHAT
input yf1 yf2 uf1 uf2 a1 a2 a3 a4
output yh
yh=(yf1 * a1)+(yf2 * a2)+(uf1 * a3) + (uf2 * a4)
end
```

Input Signal Generator for Phase 1 Testing

```
Continuous System YBLD
input v w x y
output yh
yh=(v*a1)+(w*a2)+(x*a3)+y*a4)
a1:1
a2:2
a3:3
a4:2
end
```

Phase 1 Least Squares Estimation Testing Connections

```
Connecting System CON
time t
v[ybld]=x[sins]
w[ybld]=y[sins]
x[ybld]=z[sins]
y[ybld]=xx[sins]
uf1[pdot]=z[sins]
uf2[pdot]=xx[sins]
yf1[pdot]=x[sins]
yf2[pdot]=y[sins]
uf1[ahat]=z[sins]
uf2[ahat]=xx[sins]
yf1[ahat]=x[sins]
yf2[ahat]=y[sins]
y[ahat]=yh[ybld]
yh[ahat]=yh[yhat]
```

```
p1[ahat]=pd1[pdot]
p2[ahat]=pd2[pdot]
p3[ahat]=pd3[pdot]
p4[ahat]=pd4[pdot]
p5[ahat]=pd5[pdot]
p6[ahat]=pd6[pdot]
p7[ahat]=pd7[pdot]
p8[ahat]=pd8[pdot]
p9[ahat]=pd9[pdot]
p10[ahat]=pd10[pdot]
p11[ahat]=pd11[pdot]
p12[ahat]=pd12[pdot]
p13[ahat]=pd13[pdot]
p14[ahat]=pd14[pdot]
p15[ahat]=pd15[pdot]
p16[ahat]=pd16[pdot]
uf1[yhat]=z[sins]
uf2[yhat]=xx[sins]
yf1[yhat]=x[sins]
yf2[yhat]=y[sins]
a1[yhat]=ah1[ahat]
a2[yhat]=ah2[ahat]
a3[yhat]=ah3[ahat]
a4[yhat]=ah4[ahat]
end
```

Third-Order Estimator Input Prefilter "UFn" (figure 13)

```
Continuous System UFn
input u
output uf
state x1 x2 x3
der x1d x2d x3d
a1=-1*3.0*lamda
a2=-1*3.0*lamda*lamda
a3=-1*lamda*lamda*lamda
c3=lamda*lamda*lamda
uf=c1 * x1 + c2 * x2 + c3 * x3
x1d=a1 * x1 + a2 * x2 + a3 * x3 + u
x2d=a4 * x1 + a5 * x2 + a6 * x3
x3d=a7 * x1 + a8 * x2 + a9 * x3
a4:1.0
a5:0.0
a6:0.0
a7:0.0
a8:1.0
a9:0.0
c1:0.0
c2:0.0
lamda:100.0
end
```

Third-Order Estimator Output Prefilter "YFn" (figure 13)

```
Continuous System YFn
input y
output yf
state x1 x2 x3
der x1d x2d x3d
a1=-1*3.0*lamda
a2=-1*3.0*lamda*lamda
a3=-1*lamda*lamda*lamda
c3=lamda*lamda*lamda
yf=c1 * x1 + c2 * x2 + c3 * x3
x1d=a1 * x1 + a2 * x2 + a3 * x3 + y
x2d=a4 * x1 + a5 * x2 + a6 * x3
x3d=a7 * x1 + a8 * x2 + a9 * x3
a4:1.0
a5:0.0
a6:0.0
a7:0.0
a8:1.0
a9:0.0
c1:0.0
c2:0.0
lamda:100.0
end
```

Third-Oder Estimator Input Prefilter "YT" (figure 13)

```
Continuous System YT
input u
output y
state x1 x2 x3
der x1d x2d x3d
a0=2*k+bb/aa
d=1.0
c1=(a0-3*lf)
c2=0.0
c3=0.0
a1=-1*3.0*lf
a2=-1*3.0*lf*lf
a3=-1*lf*lf*lf
y=c1 * x1 + c2 * x2 + c3 * x3 + d * u
x1d=a1 * x1 + a2 * x2 + a3 * x3 + u
x2d=a4 * x1 + a5 * x2 + a6 * x3
x3d=a7 * x1 + a8 * x2 + a9 * x3
a4:1.0
a5:0.0
a6:0.0
a7:0.0
a8:1.0
a9:0.0
c1:0.0
c2:0.0
```

```
lf:100.0
k:100.0
aa:200.0
bb:2050.0
end
```

Third-Order Input Signal Generator for Phase 2 Testing

```
Continuous System ABCD
input u
output y
state x1 x2 x3
der x1d x2d x3d
a1=-1*(2*k+b/a)
a2=-1*(2*k*b/a) + k*k + k*k*c/a)
a3=-1*(k*k*b/a + k*k*d/a)
c2=2*k*k*c/(q*a)
c3=2*k*k*d/(q*a)
x1d=a1*x1 + a2*x2 + a3*x3 + u
x2d=a4*x1 + a5*x2 + a6*x3
x3d=a7*x1 + a8*x2 + a9*x3
y=c1*x1 + c2*x2 + c3*x3
a4:1.0
a5:0.0
a6:0.0
a7:0.0
a8:1.0
a9:0.0
c1:0.0
a:200.0
b:2050.0
c:200.0
d:1845.0
q:0.5
k:100.0
end
```

Phase 2 Least Squares Estimation Testing Connections

```
Connecting System CON
time t
sig1=x[sins]+y[sins]+z[sins]
sig2=xx[sins]+yy[sins]+zz[sins]
sig3=qq[sins]
sig4=sig1+sig2+sig3
u[abcd]=sig4
u[uf1]=sig4
u[uf2]=sig4
u[yt]=y[abcd]
y[yf1]=y[abcd]
y[yf2]=y[abcd]
uf1[pdot]=uf[uf1]
```

```
uf2[pdot]=uf[uf2]
yf1[pdot]=yf[yf1]
yf2[pdot]=yf[yf2]
uf1[ahat]=uf[uf1]
uf2[ahat]=uf[uf2]
yf1[ahat]=yf[yf1]
yf2[ahat]=yf[yf2]
y[ahat]=y[yt]
yh[ahat]=yh[yhat]
p1[ahat]=pd1[pdot]
p2[ahat]=pd2[pdot]
p3[ahat]=pd3[pdot]
p4[ahat]=pd4[pdot]
p5[ahat]=pd5[pdot]
p6[ahat]=pd6[pdot]
p7[ahat]=pd7[pdot]
p8[ahat]=pd8[pdot]
p9[ahat]=pd9[pdot]
p10[ahat]=pd10[pdot]
p11[ahat]=pd11[pdot]
p12[ahat]=pd12[pdot]
p13[ahat]=pd13[pdot]
p14[ahat]=pd14[pdot]
p15[ahat]=pd15[pdot]
p16[ahat]=pd16[pdot]
uf1[yhat]=uf[uf1]
uf2[yhat]=uf[uf2]
yf1[yhat]=yf[yf1]
yf2[yhat]=yf[yf2]
a1[yhat]=ah1[ahat]
a2[yhat]=ah2[ahat]
a3[yhat]=ah3[ahat]
a4[yhat]=ah4[ahat]
end
```

Parameter Mapping System "DCE" (figure 14)
(parameters default to cs,es until routine is triggered at time = 1.0)

```
Discrete System DCE
input t1 t2 t3 t4
output c e
state x3 x4
new nx3 nx4
time t
tsamp ts
c=x3
e=x4
h2=1/(cs*cs*l*l) + 4/(q*q*cs*cs*l*l)
b3=1 - es/(cs*l)
b4=3/l - (2*es)/(l*l*cs) - 1/l
mn3=(t4*2)/(q*cs*l*h2) - (t2-b4)/(cs*l*h2)
mn4=(t3*2)/(q*cs*l*h2) - (t1-b3)/(cs*l*h2)
nx3= if t<1.0 then cs else if t>1.01 then x3 else mn3
```

```
nx4= if t<1.0 then es else if t>1.01 then x4 else mn4
ts=t+dt
dt:0.01
q:0.5
l:100.0
cs:200.0
es:2050.0
end
```

Third-Oder Adaptive Force-Force Extender Controller "H_h" (figure 14)

```
Continuous System HH
input c e u
output y
state x1 x2 x3
der x1d x2d x3d
d=(1.0/(c*18860.0))*k*k
c1=k*k*((1.0/(c*530.52))-(1.0/(c*18860.0))*(2*k+(e/c)))
c2=k*k*((1.0/(c*11.83))-(1.0/(c*18860.0))*(k*k+2.0*(e/c)*k))
c3=k*k*((1.0/c)-0.(1.0/(c*18860.0))*(k*k*(e/c)))
a1=-1*(2*k+(e/c))
a2=-1*(k*k+2.0*(e/c)*k)
a3=-1*k*k*(e/c)
y=(2/z*c1*x1) + (2/z*c2*x2) + (2/z*c3*x3) + (2/z*d*u)
x1d=(a1 * x1) + (a2 * x2) + (a3 * x3) + u
x2d=(a4 * x1) + (a5 * x2) + (a6 * x3)
x3d=(a7 * x1) + (a8 * x2) + (a9 * x3)
a4:1.0
a5:0.0
a6:0.0
a7:0.0
a8:1.0
a9:0.0
z:0.5
k:100.0
end
```

Third-Order Adaptive Force-Force Extender Controller "H_n" (figure 14)

```
Continuous System HN
input c e u
output y
state x1 x2 x3
der x1d x2d x3d
d=(1.0/(c*18860.0))*k*k
c1=k*k*((1.0/(c*530.52))-(1.0/(c*18860.0))*(2*k+(e/c)))
c2=k*k*((1.0/(c*11.83))-(1.0/(c*18860.0))*(k*k+2.0*(e/c)*k))
c3=k*k*((1.0/c)-0.(1.0/(c*18860.0))*(k*k*(e/c)))
a1=-1*(2*k+(e/c))
a2=-1*(k*k+2.0*(e/c)*k)
a3=-1*k*k*(e/c)
```

```
y=(c1*x1) + (c2*x2) + (c3*x3) + (d*u)
x1d=(a1 * x1) + (a2 * x2) + (a3 * x3) + u
x2d=(a4 * x1) + (a5 * x2) + (a6 * x3)
x3d=(a7 * x1) + (a8 * x2) + (a9 * x3)
a4:1.0
a5:0.0
a6:0.0
a7:0.0
a8:1.0
a9:0.0
k:100.0
end
```

Phase 3 Least Squares Estimation Testing Connections

Connecting System CON

```
time t
out=y[ss]
sig1=x[sins]+y[sins]+z[sins]
sig2=xx[sins]+yy[sins]+zz[sins]
sig3=qq[sins]+ww[sins]+vv[sins]
uh=sig1+sig2
sum=y[gdh]+y[gdn]+y[gp]
u[gp]=kb*kda*(k1*(k0*(y[hh]+y[hn])=kio*y[ss])-k2*kad*kt*sum)
u[gdh]=uh-y[tk2]
u[hh]=uh=y[tk2]
c[hh]=c[dce]
e[hh]=e[dce]
u[gdn]=p-(e0*y[ss]+e1*sum)
u[hn]=p-(e0*y[ss]+e1*sum)
c[hn]=c[dce]
e[hn]=e[dce]
u[tk2]=y[ss]
u[ss]=sum
u[uf1]=uh=y[tk2]
u[uf2]=uh=y[tk2]
u[yt]=-1*(p-(e0*y[ss]+e1*sum))
y[yf1]=-1*(p-(e0*y[ss]+e1*sum))
y[yf2]=-1*(p-(e0*y[ss]+e1*sum))
uf1[pdot]=uf[uf1]
uf2[pdot]=uf[uf2]
yf1[pdot]=yf[yf1]
yf2[pdot]=yf[yf2]
uf1[ahat]=uf[uf1]
uf2[ahat]=uf[uf2]
yf1[ahat]=yf[yf1]
yf2[ahat]=yf[yf2]
y[ahat]=y[yt]
yh[ahat]=yh[yhat]
p1[ahat]=pd1[pdot]
p2[ahat]=pd2[pdot]
p3[ahat]=pd3[pdot]
p4[ahat]=pd4[pdot]
```

```
p5[ahat]=pd5[pdot]
p6[ahat]=pd6[pdot]
p7[ahat]=pd7[pdot]
p8[ahat]=pd8[pdot]
p9[ahat]=pd9[pdot]
p10[ahat]=pd10[pdot]
p11[ahat]=pd11[pdot]
p12[ahat]=pd12[pdot]
p13[ahat]=pd13[pdot]
p14[ahat]=pd14[pdot]
p15[ahat]=pd15[pdot]
p16[ahat]=pd16[pdot]
uf1[yhat]=uf[uf1]
uf2[yhat]=uf[uf2]
yf1[yhat]=yf[yf1]
yf2[yhat]=yf[yf2]
a1[yhat]=ah1[ahat]
a2[yhat]=ah2[ahat]
a3[yhat]=ah3[ahat]
a4[yhat]=ah4[ahat]
t1[dce]=ah1[ahat]
t2[dce]=ah2[ahat]
t3[dce]=ah3[ahat]
t4[dce]=ah4[ahat]
kb:0.00465
kda:0.004883
k1:0.94
k0:1592.0
kio:1592.0
k2:0.00977
kad:1638.4
kt:0.169
e0:2050.0
e1:200.0
p:0.0
end
```

Input Signal Generator for Phase 1 BGF Testing

```
Continuous System YBLD
time t
input v w x y
output yh
a2h=1.0 + sin(t/2.0)
a4h=2.0 + sin(t/2.0)
a2=if t>0.0 then a2h else a2s
a4=if t>4.0 then a4h else a4s
yh=(v*a1)+(w*a2)+(x*a3)+y*a4)
a1:1
a2s:2
a3:3
a4s:2
end
```

Third-Order Input Signal Generator for Phase 2 BGF Testing

```
Continuous System ABCD
time t
input u
output y
state x1 x2 x3
der x1d x2d x3d
dh=1845+(10.0*sin(t/2.0))
c=cs
d=if t>0.0 then dh else ds
a1=-1*(2*k+b/a)
a2=-1*(2*k*b/a) + k*k + k*k*c/a)
a3=-1*(k*k*b/a + k*k*d/a)
c2=2*k*k*c/(q*a)
c3=2*k*k*d/(q*a)
x1d=a1*x1 + a2*x2 + a3*x3 + u
x2d=a4*x1 + a5*x2 + a6*x3
x3d=a7*x1 + a8*x2 + a9*x3
y=c1*x1 + c2*x2 + c3*x3
a4:1.0
a5:0.0
a6:0.0
a7:0.0
a8:1.0
a9:0.0
c1:0.0
a:200.0
b:2050.0
cs:200.0
ds:1845.0
q:0.5
k:100.0
end
```

Fourth-Order BGF Gain Update Model "PDOT" (figure 14)

```
Continuous System PDOT
input yf1 yf2 uf1 uf2
output pd1 pd2 pd3 pd4 pd5 pd6 pd7 pd8 pd9 pd10 pd11 pd12
output pd13 pd14 pd15 pd16
state p1 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13 p14 p15 p16
der p1d p2d p3d p4d p5d p6d p7d p8d p9d p10d p11d p12d p13d
der p14d p15d p16d
sum1=p1*p1+p2*p2+p3*p3+p4*p4+p5*p5+p6*p6+p7*p7+p8*p8+p9*p9
sum2=p10*p10+p11*p11+p12*p12+p13*p13+p14*p14+p15*p15+p16*p16
sum3=sum1+sum2
norm=sqrt(sum3)
lam=l0*(1-(norm/k0)
pd1=p1
pd2=p2
```

```
pd3=p3
pd4=p4
pd5=p5
pd6=p6
pd7=p7
pd8=p8
pd9=p9
pd10=p10
pd11=p11
pd12=p12
pd13=p13
pd14=p14
pd15=p15
pd16=p16
w1=yf1*yf1
w2=yf2*yf1
w3=uf1*yf1
w4=uf2*yf1
w5=yf2*yf2
w6=uf1*yf2
w7=uf2*yf2
w8=uf1*uf1
w9=uf2*uf1
w10=uf2*uf2
i1=(p1 * w1) + (p5 * w2) + (p9 * w3) + p13 * w4)
i2=(p2 * w1) + (p6 * w2) + (p10 * w3) + p14 * w4)
i3=(p3 * w1) + (p7 * w2) + (p11 * w3) + p15 * w4)
i4=(p4 * w1) + (p8 * w2) + (p12 * w3) + p16 * w4)
i5=(p1 * w2) + (p5 * w5) + (p9 * w6) + p13 * w7)
i6=(p2 * w2) + (p6 * w5) + (p10 * w6) + p14 * w7)
i7=(p3 * w2) + (p7 * w5) + (p11 * w6) + p15 * w7)
i8=(p4 * w2) + (p8 * w5) + (p12 * w6) + p16 * w7)
i9=(p1 * w3) + (p5 * w6) + (p9 * w8) + p13 * w9)
i10=(p2 * w3) + (p6 * w6) + (p10 * w8) + p14 * w9)
i11=(p3 * w3) + (p7 * w6) + (p11 * w8) + p15 * w9)
i12=(p4 * w3) + (p8 * w6) + (p12 * w8) + p16 * w9)
i13=(p1 * w4) + (p5 * w7) + (p9 * w9) + p13 * w10)
i14=(p2 * w4) + (p6 * w7) + (p10 * w9) + p14 * w10)
i15=(p3 * w4) + (p7 * w7) + (p11 * w9) + p15 * w10)
i16=(p4 * w4) + (p8 * w7) + (p12 * w9) + p16 * w10)
p1d=(lam*p1)-((i1 * p1)+(i5 * p2)+(i9 * p3)+(i13 * p4))
p2d=(lam*p2)-((i2 * p1)+(i6 * p2)+(i10 * p3)+(i14 * p4))
p3d=(lam*p3)-((i3 * p1)+(i7 * p2)+(i11 * p3)+(i15 * p4))
p4d=(lam*p4)-((i4 * p1)+(i8 * p2)+(i12 * p3)+(i16 * p4))
p5d=(lam*p5)-((i1 * p5)+(i5 * p6)+(i9 * p7)+(i13 * p8))
p6d=(lam*p6)-((i2 * p5)+(i6 * p6)+(i10 * p7)+(i14 * p8))
p7d=(lam*p7)-((i3 * p5)+(i7 * p6)+(i11 * p7)+(i15 * p8))
p8d=(lam*p8)-((i4 * p5)+(i8 * p6)+(i12 * p7)+(i16 * p8))
p9d=(lam*p9)-((i1 * p9)+(i5 * p10)+(i9 * p11)+(i13 * p12))
p10d=(lam*p10)-((i2 * p9)+(i6 * p10)+(i10 * p11)+(i14 * p12))
p11d=(lam*p11)-((i3 * p9)+(i7 * p10)+(i11 * p11)+(i15 * p12))
p12d=(lam*p12)-((i4 * p9)+(i8 * p10)+(i12 * p11)+(i16 * p12))
p13d=(lam*p13)-((i1 * p13)+(i5 * p14)+(i9 * p15)+(i13 * p16))
p14d=(lam*p14)-((i2 * p13)+(i6 * p14)+(i10 * p15)+(i14 *p16))
```

```
p15d=(lam*p15)-((i3 * p13)+(i7 * p14)+(i11 * p15)+(i15 *p16))
p16d=(lam*p16)-((i4 * p13)+(i8 * p14)+(i12 * p15)+(i16 *p16))
k0:12000.0
l0:16.0
end
```

Second-Order Multi-Input Gain Update Model "PDOT"

```
Continuous System PDOT
input w1 w2
output pd1 pd2 pd3 pd4
state p1 p2 p3 p4
der p1d p2d p3d p4d
sum1=p1*p1+p2*p2+p3*p3+p4*p4
norm=sqrt(sum1)
lam=l0*1-(norm/k0))
pd1=p1
pd2=p2
pd3=p3
pd4=p4
j1=w1*w1
j2=w1*w2
j3=w2*w2
i1=(p1*j1)+(p3*j2)
i2=(p2*j1)+(p4*j2)
i3=(p1*j2)+(p3*j3)
i4=*p2*j2)+(p4*j3)
p1d=(lam*p1)-((i1*p1)+(i3*p2))
p2d=(lam*p2)-((i2*p1)+(i4*p2))
p3d=(lam*p3)-((i1*p3)+(i3*p4))
p4d=(lam*p4)-((i2*p3)+(i4*p4))
k0:12000.0
l0:16.0
end
```

Second-Order Multi-Input Parameter Update Model "AHAT"

```
Continuous System AHAT
input w1 w2 y yh p1 p2 p3 p4
output ah1 ah2
state a1 a2
der a1d a2d
e1=yh-y
ah1=a1
ah2=a2
a1d=-1*e1*((w1*p1)+(w2*p2))
a2d=-1*e1*((w1*p3)+w2*p4))
end
```

Multi Input Parameter Error Update Model "YHAT"

```
Continuous System YHAT
input w1 w2 a1 a2
output yh
yh=(w1*a1)+(w2*a2)
end
```

Multi Input BGF Estimation Testing Connections

```
Connecting System CON
time t
out=y[ss]
e0h = 1845.0+(10.0*sin(t/2.0))
e1h = 200.0+(10.0*sin(t/2.0))
e0=if t>0.0 then e0h else e0s
e1=if t>10.0 then e1h else e1s
sig1=x[sins]+y[sins]+z[sins]
sig2=xx[sins]+yy[sins]+zz[sins]
sig3=qq[sins]+ww[sins]+vv[sins]
uh=sig1+sig2
sum=y[gdh]+y[gdn]+y[gp]
u[gp]=kb*kda*(k1*(k0*(y[hh]+y[hn])=kio*y[ss])-k2*kad*kt*sum)
u[gdh]=uh-y[tk2]
u[hh]=uh=y[tk2]
c[hh]=ah2[ahat]
e[hh]=ah1[ahat]
u[gdn]=p-(e0*y[ss]+e1*sum)
u[hn]=p-(e0*y[ss]+e1*sum)
c[hn]=ah2[ahat]
e[hn]=ah1[ahat]
u[tk2]=y[ss]
u[ss]=sum
w1[pdot]=y[ss]
w2[pdot]=sum
w1[ahat]=y[ss]
w2[ahat]=sum
y[ahat]=-1*(p-(e0*y[ss]+e1*sum))
yh[ahat]=yh[yhat]
p1[ahat]=pd1[pdot]
p2[ahat]=pd2[pdot]
p3[ahat]=pd3[pdot]
p4[ahat]=pd4[pdot]
w1[yhat]=y[ss]
w2[yhat]=sum
a1[yhat]=ah1[ahat]
a2[yhat]=ah2[ahat]
kb:0.00465
kda:0.004883
k1:0.94
k0:1592.0
kio:1592.0
k2:0.00977
```

```
kad:1638.4
kt:0.169
e0s:2050.0
e1s:200.0
p:0.0
end
```

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
Chief of Naval Operations (NOP-987)	1
Chief of Naval Research (OCNR-4520, OCNR-4525)	2
Defense Technical Information Center	12
Center for Naval Analyses	1